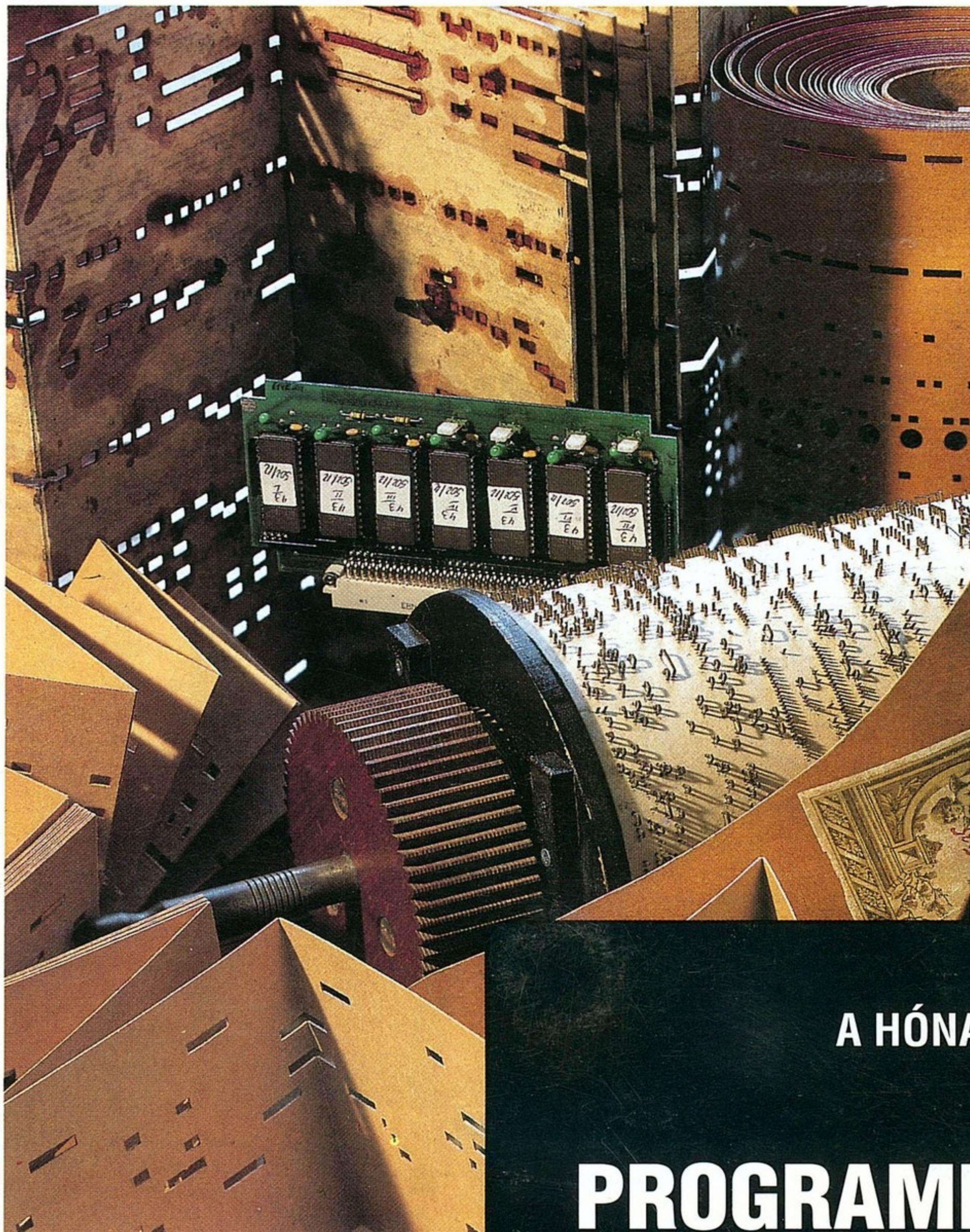
1995 / MÁJUS ÁRA: 297 FT

ÚJ ALAPLAP

SZÁMÍTÁSTECHNIKAI MAGAZIN MÁGNESLEMEZ MELLÉKLETTEL



Ikonológia

Kedit for Windows

A játékpiac nem játék

„Pascal-trilógia"

Ötéves az Ifabo

A HÓNAP TÉMÁJA:

PROGRAMNYELVÚJÍTÁS

Nincs kockázat, van RISC!

Az archívkészítők új hulláma

Miért tévedett a Pentium?

A LEMEZEN:

Postai adatbázis
Mit tud a Multikey?
Matematikai alapépítés
Új csillag, a RAR
3D társasjáték

Megújult remekmű a piacvezetőtől.

NetWare 4.1



A **NetWare**-t, a piacvezető hálózati operációs rendszert emberek és cégek közötti kommunikációra tervezték. Döntő fontosságú fejlesztések biztosítják, hogy milliók kommunikálhassanak segítségével.

A felhasználóbarát **NetWare 4.1** mindenféle méretű cég hálózatának teljesítményét megnöveli. Mi ezt egyszerűen "átható számítástechnikának" hívjuk.

Lényegesen leegyszerűsíti a **NetWare 4.1** a PC-k előtt töltött időt: egylépéses bejelentkezéssel hozzáférhetünk az összes kiszolgálóhoz és egyéb erőforráshoz. Játszi könnyedséggel navigálhatunk a hálózaton az új, grafikus felhasználói felületnek köszönhetően. Az optimális felügyeleti rendszerrel pedig nyomon követhetünk minden tevékenységet.

A NetWare 4.1-ből származó előnyök:

- **Kényelmesebb** – az új nyomtatókezelés miatt
- **Egyszerűbb** – a grafikus felhasználói felület miatt
- **Rugalmasabb** – ideális bármilyen méretű cég számára
- **Hatékonyabb** – a továbbfejlesztett felügyeletnek köszönhetően
- **Gazdaságosabb** – a hardver optimális kihasználása miatt

Megnöveli a memóriát a továbbfejlesztett adattömörítő technológia, az egyszerű nyomtatókezelés pedig gyors és hatékony nyomtatást – a hardver optimális kihasználását – teszi lehetővé.

Elismerten vetélytárs nélküliek a **NetWare 4.1** által használt biztonsági szabványok. Minden szinthez, erőforráshoz és adathoz való hozzáférést hitelesítenek és naplóznak az új vezérlőfunkciók. Hatékonyan elkerülhető a hardvermeghibásodásból származó adatvesztés is.

Hamar észrevehető, hogyan újult meg a klasszikus szoftver a kommunikáció segítése érdekében. Eljött a **NetWare 4.1** ideje.

Bármilyen kérdésével forduljon bizalommal a Novell hivatalos viszonteladóihoz.

Érdeklődjön az alábbi fax számon: 266-6360

NOVELL®

INFORMÁCIÓKÉRÉS: 0536 ◄

ÚJ ALAPLAP

A Mikroszámítógép Magazin és az Alaplap hagyományait folytató számítástechnikai folyóirat

Megjelenik havonta, mágneslemez melléklettel

Főszerkesztő:
Faklen Pál

Főszerkesztő-helyettes:
Varga János

Szerkesztő:
Jakab Ágnes

A szerkesztőbizottság tagjai:
Barna László, Brüll Károly, Csórián Sándor, Feleki Zoltán, Herczeg József, Horlai János, Jánosi Tibor, Kis János, Nagy Gábor, Sík Zoltán, Szondi Egon János, Vargha Dénes, Vékony Tamás, Villányi László

Szerkesztőség és kiadó:
1538 Budapest I., Márvány u. 17.
Telefon: 156-3211 / 200, 214
Fax (manuális): 156-3211 / 201

Felelős kiadó:
Faklen Pál

Terjesztés:
Megyes Zsuzsa

Hirdetésszervezés:
Árvai Katalin, Bogácsi Mária, Tóth Zoltán

Külföldi hirdetések:
PubliCity
Reklám- és Médiaügynökség
1537 Budapest I., Márvány u. 17.
Telefon: 156-1182 Fax: 175-3539

Példányszámadatok hitelesítése:
Magyar Terjesztésellenőrző Szövetség
MATESZ
és Price Waterhouse

Nyomtatás:
Zalai Nyomda Rt, Zalaegerszeg
Felelős vezető:
Somogyi Tibor ügyvezető igazgató

Terjeszti:
A Magyar Posta Rt, a Nemzeti Hírlapkereskedelmi Rt, a Hírker Rt, az Extra-Hír Rt, számos számítástechnikai szaküzlet és más terjesztő

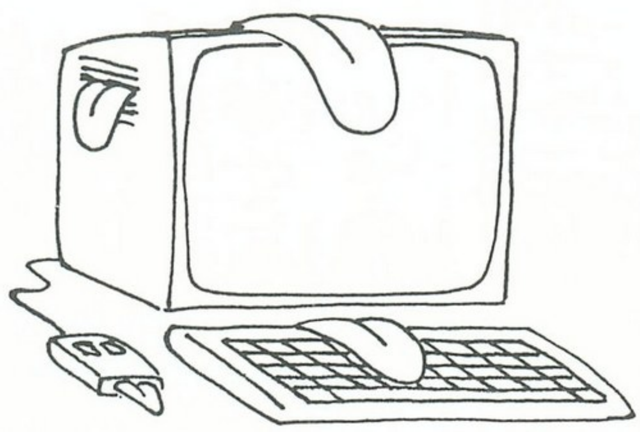
Előfizethető a kiadónál:
Új Alaplap Kiadói Kft,
1538 Budapest, Pf. 571
Átutalás: Agrobank 219-93789/ 10878060-70030011
Példányonkénti ár: 297 Ft
Évi előfizetési díj: 2970 Ft

Külföldre terjeszti a Kultúra, H-1389 Budapest, Pf. 149

HU ISSN 1217-7598

A HÓNAP TÉMÁJA: PROGRAMNYELVÚJÍTÁS

(Összeállította: Sziebig Andrea)



SZOFTVERPORTÉKA

GÉPRAJZ

UNIXUMOK

PRO DOMO

KIRAKAT

FOGÓDZÓ

KÖZKINCS

OKTATÁS

MŰHELY

TUDÁSTECHNOLÓGIA

PROGRAMOZÁSTECHNIKA

KÖNYVESPOLC

PALETTA

MÁGNESLEMEZ MELLÉKLET ❑

Feleki Zoltán karikatúrái

Címlapképünk archív felvétel

FLOPPYLEMEZ-VÁSÁR

MIC®

FLOPPY DISKETTES FROM THE U.S.A.

MINDEN EGYES DISZKET MEGVIZSGÁLUNK
ÉS MÉRÜNK,
HOGY 100%-IG HIBAMENTES LEGYEN!

3,5" HD

MIC, JVC, KAO	96 Ft + áfától
– ipari csomagolásban	74 Ft + áfától

3,5" DD

MIC és ipari	36 Ft + áfától

5,25" HD

MIC, JVC, KAO	60 Ft + áfától
– ipari csomagolásban (színes)	48 Ft + áfától

5,25" DD

– ipari csomagolásban	28 Ft + áfától

AZ IFABO ALATT 25% KEDVEZMÉNY!

Diákoknak, iskoláknak, egészségügyi intézményeknek 20% kedvezmény.
Viszonteladói és mennyiségi kedvezmények!
Szállítás raktárról, azonnal. Rendelésfelvétel telefonon és faxon is.
Turbó floppy-másolás: 3,5"-es HD, 3000 db/nap.

Szoftvermásolás, diszkettmásolás
Ipari CD-író, szoftvermásoló berendezések értékesítése, lízingje

Gyártó – Importőr:

SOUL EUROPE CO. HUNGARY
1089 Budapest, Győrffy I. u. 1.
Tel.: 186-2713, 113-5605 Fax: 186-2713

TETA MAGNETIC KFT.
TETA MANAGER SHOP
1134 Budapest, Váci út 19. Tel./Fax: 111-5004

INFORMÁCIÓKÉRÉS: 0563 ▶

GEMOFIS KFT.

Budapest XIV. Hungária krt. 131.
Tel / Fax: **12-11-539**
GSM: (06-30)-**428-132**

AutoCAD LT for Windows	55.000
AutoCAD R12 magyar Akció!	**198.000**
AutoCAD R12 - R13 upgr. angol Akció!	**51.000**
Borland C++ 4.5	64.760 / 27.410
Paradox for Windows 5.0 / 4.5	31.270 / 14.100
DBase 5 DOS vagy Windows, promo!	19.300
Clipper 5.2 D USA	27.800
Clipper 5.2 D+Exosp.+dBfast / CaTools	36.040
Corel CD Creator (CD író software)	28.400
Corel Draw 5 CD spec / upgr.4	55.660 / 30.190
Corel 3 spec. / Corel 4 spec.	14.000 / 19.900
IBM OS/2 3.0 War+Win CD	20.380
IBM OS/2 3.0 Warp CD	**12.000 / 8.380**
MS Access 2.0 angol vagy magyar /upgr.	43.240 / 16.750
MS DOS 6.22	6.840
MS Windows 3.1 Hun vagy EE	13.180
MS Windows 3.11 for workgroups EE	13.180
MS Foxpro 2.6 Win vagy DOS / upgr.	43.240 / 16.780
MS Office 4.2 magyar / angol	55.000 / 59.500
MS Office 4.3 magyar vagy angol	**75.000**
MS Visual Basic 3.0	17.470 / 6.380
MS Works 3.0 magyar	12.190
Novell DOS 7	12.000
Norton Commander 5.0	**11.550**
Norton Utilities 8.0 spec.	**15.000**
Quatro Pro 6.0 / Quatro 5.0	42.110 / 8.120
Watcom C/C++ 10.0 (32 bit) CD	34.850
HP LaserJet 4L / DeskJet 520	**90.640 / 44.820**

Részletes listánkért hívja a FAXBANK-ot: 180-8611, kód: 1476 # Ton üzemmódban.

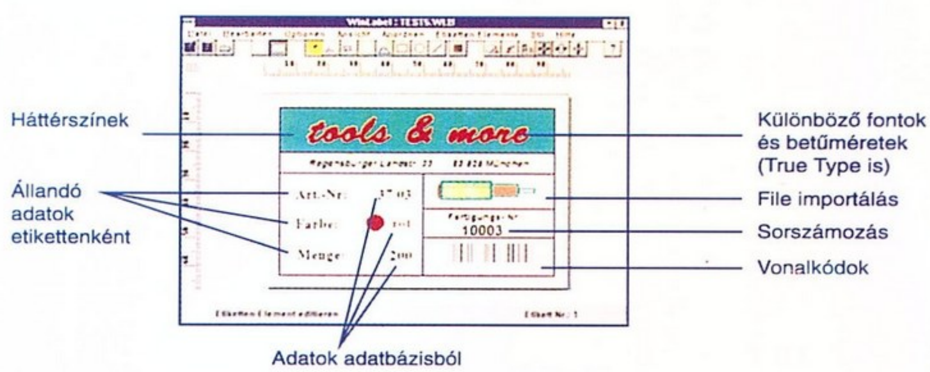
Az árak készpénz fizetésre vonatkoznak és nem tartalmazzák az ÁFÁ-t. Az árváltoztatás jogát fenntartjuk!

INFORMÁCIÓKÉRÉS: 0524 ▶

Zweckform

WinLabel®

ETIKETTFELIRATOZÓ SZOFTVER magyar változatban a WINLABEL 1.1



LÉZER – INKJET – MÁTRIX NYOMTATÓHOZ

• Sorszámozás • Grafika • Adatbázis • Vonalkódok •
• Szövegszerkesztés •

DE A LÉNYEG: A PROGRAMHOZ TARTOZÓ VALAMENNYI ETIKETT ÁLLANDÓAN KAPHATÓ!

ÁRA CSAK 6800 Ft + áfa

ARECO INFORMATIKAI KFT.

Üzlet: Budapest VI., Podmaniczky u. 9.

Telefon: 112-5084, 111-6802, 111-1456 Telefax: 131-0340
Nyitva tartás: hétfőtől–péntekig 8-tól 18 óráig
Csomagküldés utánvéttel

Vásárlás esetén ez a kupon 5% kedvezményt ér Önnek!

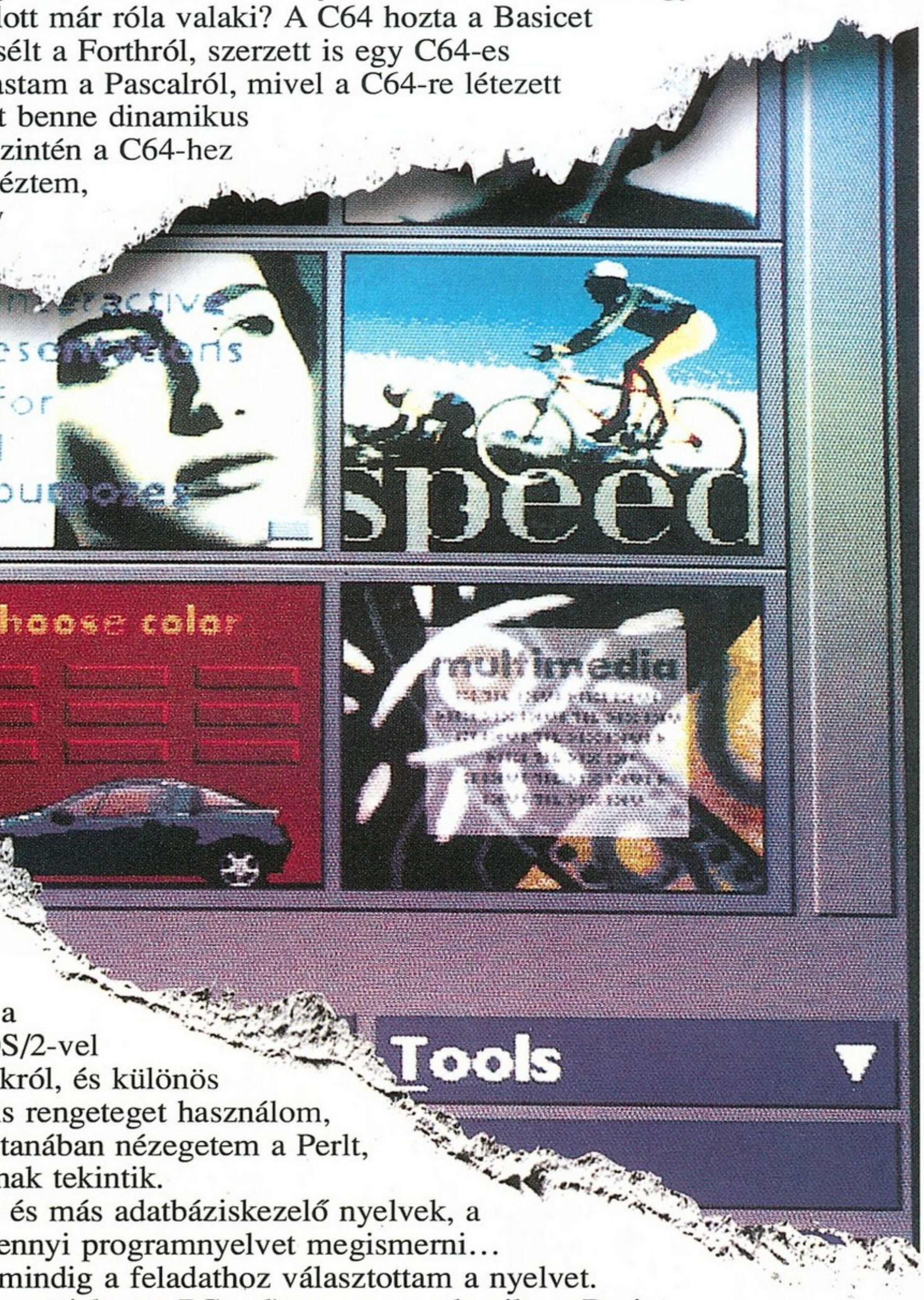
INFORMÁCIÓKÉRÉS: 0503 ▶

Nyelvemlékeim

Hogy mi adja témánk aktualitását? Az, hogy mindig aktuális. Bevezető gyanánt ez alkalommal egyik szerzőnknek, Horlai Jánosnak a témához fűződő kapcsolatáról adott szubjektív vallomását adjuk közre.

„A 70-es évek elején, a középiskolában kezdtem számítástechnikát tanulni. A nyelv, amelyet akkor megismertünk, természetesen a Fortran volt. Iskolánk egy TPA géppel rendelkezett, amelynek operációs rendszerét induláskor lyukszalagról töltöttük be. Ennek nyelvét, ha jól emlékszem, Fokalnak hívták, megtanultuk ezt is. Közben egy nyári munka során megismertem egy régi szovjet számítógépet, és annak gépközeli nyelve, a Mitra elsajátítása töltötte ki a munkaidőmet.

Első munkahelyemen a Cobolt kellett használnom, bár volt ott is Fortran, egy fejlett, sokat tudó változat. Nemsokára — szükségből — következett a Honeywell assemblere, amelyet GMAP-nek hívtak. Egy barátom pedig megismertetett a CDL2-vel — hallott már róla valaki? A C64 hozta a Basicet és a C64 assemblerét. Előbb említett barátom mesélt a Forthról, szerzett is egy C64-es verziót, hetekig próbálgattuk. Akkoriban már olvastam a Pascalról, mivel a C64-re létezett egy erősen korlátozott változata (például nem volt benne dinamikus tárkezelés), megtanultam, sőt használtam azt is. Szintén a C64-hez lehetett szerezni egy Logo-változatot, azt is megnéztem, nagyon tetszett. Közben megjelent a C-könyv, így valamennyire megtanultam a C-t könyvből… azóta sem tudok C-ben programozni.



A PC megjelenése után megszállott pascalosként elkezdtem a Turbo Pascalt, és sok-sok évig dolgoztam benne. Mindig érdekeltek a programnyelvek, hát szereztem egy Lisp-interpretert, és jött egy kis lispezés (Xlisp-nek hívták, ha igaz). Ennek egy változata volt a Scheme, amelyhez még rendes kézikönyv is tartozott. A Prolog számított a Lisp konkurensének, így azt is megnézegettem. Véletlenül akadtam rá a Snobolra, és örömmel tanultam meg. Tudtam, hogy van folytatása, így megvettem az Icont is. Mivel programozói editorom a Kedit, annak a makrónyelve egy Rexx-variáció, hát barátkoztam a Rexxszel is. (Mellesleg ez a nyelv visszatért az OS/2-vel és a PC-DOS 7-tel.) Közben már olvastam az Awkról, és különös szerencse folytán kaptam belőle egy verziót. Ma is rengeteget használom, s ha van kedvencem, akkor az talán az Awk. Mostanában nézegetem a Perlt, amely ma divatos nyelv, sokan az Awk folytatásának tekintik.

A lista így is hosszú, pedig nem teljes, a dBase és más adatbáziskezelő nyelvek, a script és batch nyelvek mind kimaradtak. Jó volt ennyi programnyelvet megismerni… Egyrészt érdekesek, másrészt amennyire lehetett, mindig a feladathoz választottam a nyelvet. Aki ma találkozik a számítógéppel (és ez szinte egyet jelent a PC-vel), az vagy valamilyen Basic-, vagy valamilyen C-variációval ismerkedhet meg, s általában a visual jelző áll előttük). Szép, hogy a C ennyire egyeduralkodó lett (tényleg mindent meg lehet benne csinálni), de nem lenne baj, ha ma is több nyelvet választanánk, ha az emberek fejében egy feladat megoldására lehetőségként sokféle programnyelv merülne fel. Mindegyiknek más a logikája, más a szintaxisa, másban erős. Az én felfogásom szerint a programnyelvek világában is a változatosság gyönyörködtet!"

Két szomszédvár: Fortran és Algol

Rivális családalapítók

Ma már elképzelni is alig tudjuk, milyen nehéz lehetett a számítógépekkel kommunikálni, amíg csak saját speciális gépi nyelvükön adhattunk nekik utasításokat. Később azonban a számítógépeket képessé tették az emberhez közelebb álló nyelv(ek)en megfogalmazott parancsok értelmezésére.

A számítógépes programnyelvek problematikájának felbukkanása az 50-es évek derekán szoros összefüggésben volt a gépi fordítás illúziójának feltámadásával. 1949-ben Warren Weaver érvekkel támasztotta alá a gépi fordítás elvi lehetőségét, és módszereket ajánlott gyakorlati megvalósításához. Elsősorban matematikai szövegek fordítását, valamint a mikro-szövegkörnyezetek határainak és belső szerkezetének feltárását tekintette fontosnak. Vette a lapot az IBM is, és 1954-ben bemutatót tartottak az oroszról angolra fordítás lehetőségeiről. Ugyanakkor kutatások indultak a programozás automatizálásával kapcsolatban. A szakértők szerint a különböző típusú gépekre egy közös nyelvet kellett volna kidolgozni, amelyet mindegyik gép a maga módján képes megérteni.

A tojáshéj

Az első igazi programnyelvnek, a Fortrannak a gyökerei részben Európába nyúlnak. A manchesteri egyetemen Brooker autokódokat készített az egyetem Ferranti gépeire. A Mercury autokódjának sikerült először megoldania a formulák fordításának problémáját: a Mercury gép 1024 negyvenbites szót tartalmazó operatív memóriájában Brooker már olyan autokódot hozott létre, amely összetettebb algebrai formulákkal is elbánt.

A Fortranon megmaradt a Mercury tojáshéja, amelyet a változók elnevezési szokásai is tükröznek. Brooker autokódja használta először, hogy az ábécé elején és végén lévő betűk a közönséges vagy tömbváltozók, az i-től kezdődő betűk pedig az indexek vagy futóváltozók. A Mercury autokódja már ismerte a feltételes aritmetikai kifejezéseket, használt ciklusokat, és alkalmas volt hatékony tömbműveletek végzésére, sőt még a szubrutincsírák is megvoltak benne.

Később Brookernak nagy szerepe lett a gépi nyelvészet céljait szolgáló Comit nyelv kidolgozásában, amely a kifejezéseknek, a környezetfüggetlen grammatikával megfogalmazott nyelvek kifejezéseinek fordításában volt erős. Ennek alapján készítette el — három hónap (!) alatt — a Fortran II fordítóprogramját.

Megszületik a kicsi Fortran

A Fortrant tekintik az első igazi programnyelvnek. Két IBM 704-es gépre Ridgway már 1957-ben elkészítette az első fordítóprogramot, majd több más IBM gépre is sikerült átvinnie a fordítórendszert. A Fortran első változata még elég halványra sikerült, egységes koncepció helyett egyedi megoldásokkal operált, és nem ismerte a szubrutinokat sem. Először a Burroughsnak sikerült megtörnie az IBM monopóliumát, majd a többi nagy cégnek is, de nekik már a javított változat, a Fortran II gépre vitelével.

Az elismerés mellett sok kritika is érte a Fortrant: nem tisztázták kellőképpen matematikai hátterét és módszereit; megoldásai esetlegesek voltak. Mivel még pontos leírása sem volt a programozás eszperantójaként propagált nyelvnek, minden újabb implementációban másképp értelmezték a megoldandó feladatokat.

A büszke nő

Európa akkoriban még nem volt igazán lemaradva a számítástechnikában, különösen matematikai alapjainak megteremtésében. Az európai műhelyek egyrészt nem kötődtek úgy cégérdekekhez, mint a legtöbb amerikai kutatóközpont, másrészt elégedetlenkedtek az IBM titkolódzása miatt, és nemzetközi nyilvánosságot sürgettek a transzlátorkészítés elveinek és módszereinek tisztázására. Törekvéseiket több amerikai kutató is támogatta, így például John McCarthy, aki nagyjából ugyanekkor fejlesztett ki — elsősorban a mesterséges intelligencia kutatására — egy egészen más koncepciójú nyelvet, a Lispet.

Az UNESCO magára vállalta a tudományos világ összefogását egy jobban megalapozott nemzetközi programnyelv kifejlesztésére. A nyelv koncepciójának és pontos definíciójának kidolgozására sikerült összetoborozni a legkiválóbb európai és amerikai szakembereket. Az új nyelv az Algol nevet kapta. A másféle megközelítés már a névadásban is tükröződött: a formulák fordításának technikai eszköze helyett



(FORmula TRANslation) algoritmuskészítő nyelv (ALGOrithmic Language) létrehozása volt a cél.

Az első nem hivatalos ismertetés 1958-ban jelent meg az új nyelvről, de nem aratott osztatlan sikert. A kísérleti leírás hiányosságait maguk a szerzők is érezték, ezért John Backus és Peter Naur nekilátott a metanyelvi formalizmus pontosabb kidolgozásának.

Ezerkilencszázhatvan

1960-ban jelent meg az Algol-60 nyelv hivatalos leírása, amely annyira jól sikerült, hogy később a Fortran készítői is ezt tekintették példának, és a lényegesen javított Fortran IV-et már ennek mintájára definiálták 1962-ben. (A Fortran III munkaanyagként süllyedt el a feledés homályában.)

Az Algol legfontosabb fogalma a hierarchikus szervezésű blokkstruktúra volt. Pontosan definiálták az azonosítók hatásköri problémáit, a lokális és globális változók fogalmát és használatát, a láthatóságot stb. Nagy ügyetlenség volt azonban, hogy teljesen kihagyták a nyelvből az I/O definiálást, és kiirtották a pointereket.

Úgy gondolták, hogy a programozót ne érdekelje, melyik változó hol realizálódik a gépben. Így azonban lehetetlenné vált bonyolultabb struktúrák elegáns kezelése. Hasonlóan szerencsétlen megoldásnak tűnt, hogy bevezették az own deklarációt.

Az Algol-60 sem sikeredett tökéletesre, azonban nagy lépést jelentett a programozás fontos elvi és gyakorlati kérdéseinek tisztázásában. Mindezek a kérdések előzőleg vagy fel sem merültek, vagy elsikkadtak a Fortran ad hoc megoldásainak szénakazlában.

Az Algol—Fortran háború

Az új nyelv megjelenése után elkezdődött a transzlátorok készítése elsősorban Univac, CDC, Siemens, Bull és English Electric gépekre. (Az IBM hűvös tartózkodással fogadta a versenytárs felbukkanását.) Sokak számára meglepő volt, hogy az átlagosan 30 ezer soros Fortran-transzlátorok után a világosabb szerkezetű Algol fordítását feleakkora vagy annál is kisebb transzlátorokkal meg lehetett oldani; igaz, nemegyszer a nyelv valamelyes egyszerűsítésével.

A nagyobb programrendszerek azonban továbbra is inkább Fortranban készültek. Különösen igaz ez Amerikára, ahol soha nem sikerült az Algolnak igazán teret nyernie. Kellemesebbé tette a Fortrant a modulokra szabdalás, a darabonkénti fordítás lehetősége, és az egyre bővülő programozói környezet. A bájtszervezésű, integrált áramkörű gépek viharos terjedésével az Algol elvesztette addig megszerzett hadállásait is. A nagy cégek pedig egyre jobb Fortran-fordítókat készítettek.

Magyarországon a reveláció erejével hatott, amikor a nyugati nagygépek megjelenésével programozóink felfedezték, hogy nem is olyan rossz nyelv a Fortran, mint amilyen a híre. Saját bőrükön tapasztalták, hogy fordítóprogramjait össze sem lehet hasonlítani a nyugati cégektől beszerezhető Algol fordítóprogramokkal. Jó Algol-transzlátor tulajdonképpen csak akkor került hozzánk, amikor Dahl Magyarországon járt, és magával hozta a Simula-67 nyelven írt programok fordítására kidolgozott kitűnő transzlátorát. A Dahl készítette Simula-67 nyelv ugyanis korrekt kiterjesztése volt az Algolnak...

Az Algol utótörténete

Az Algol-bizottság még hosszú évekig folytatta munkáját. Látták, hogy a jó programnyelv kidolgozásához még csak az első lépéseket tették meg az Algol-60 megteremtésével, de a továbblépést illetően két táborra szakadtak. Az egyik irányzat egy teljesen univerzális nyelvet igyekezett kidolgozni, beleértve a párhuzamos programozástól a memóriakezelés megszervezéséig az égvilágon mindent.

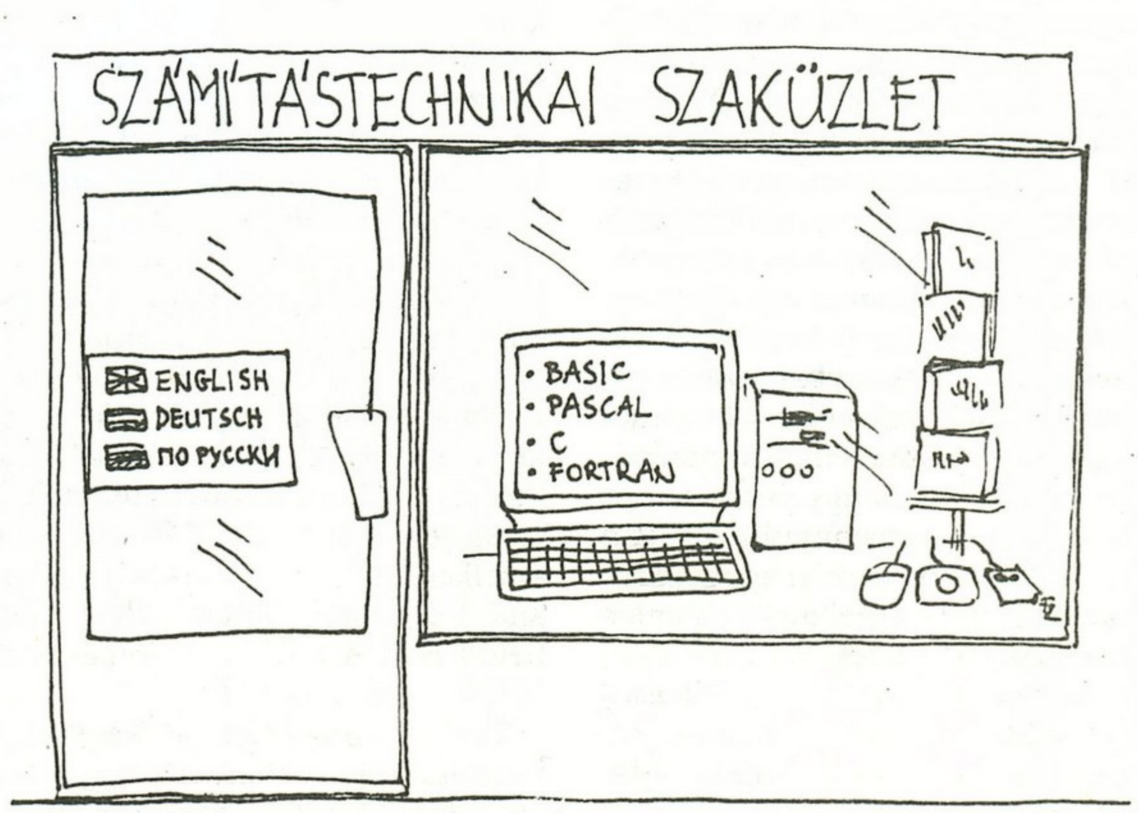
A másik irányzat a gyakorlati, programozói igények kielégítését helyezte előtérbe, és nem akarta mindenestül átvállalni az architektúra megszervezésével kapcsolatos tennivalókat. A hardver viharos fejlődését látva szükségtelennek tartották az Algol-60 teljes átstrukturálását, megelégedtek volna az ügyetlen megoldások kijavításával, a feleslegesnek ítélt dolgok elhagyásával, és azoknak a lehetőségeknek a beépítésével, amelyek kihagyása az Algol-60-ból hibás lépésnek bizonyult.

A két tábor között végül kenyértörésre került sor: a gyakorlatias irányzat kivált a bizottságból, élükön a Pascal későbbi megteremtőjével, Niklaus Wirth professzorral. Az Algol zászlaja alatt maradtak Van Wijngaarden hívei, és nekiláttak az Algol-68 néven ismertté vált elméleti konstrukció kidolgozásának, amely azonban soha nem tudott igazán elterjedni. Készült ugyan rá egy-két fordítóprogram, de azoknak is inkább csak elméleti jelentőségük volt.

Az Algol szép lassan kiöregedett, elavult. Hagyományai azonban kitörölhetetlenül beépültek a számítástechnikába. Fogalmai, struktúrájának előremutató vonásai rövidesen más nyelvekben éledtek újjá. Az Algol így nemzetségalapító ősévé vált az Algol-szerű nyelveknek (Jovial, Clip, Neliac, Mary, Simula, Pascal, Modula stb.), de lényeges dolgokat merített belőle a PL/1 és az Ada is.

A Fortrannak más lett a sorsa. Főnixmadárként többször újjáéledt, először mint Fortran-66, majd kibővítve Fortran-77-ként, legújabban pedig Fortran-90-ként került ismét előtérbe. Szívósságát időközben már az IBM is megelégelte, hiszen éppen a programozók Fortranhoz való ragaszkodása vált a PL/1 elterjedésének fő akadályává...

Vargha Dénes



Cobol, PL/1

Európai ősök nyomában

A magasszintű programnyelvek megjelenésével a különböző szakmák képviselői is „magasról", egészében szemlélték a témakörükben felmerülő feladatokat. Eltöprengtek azon, hogy milyen konkrét eszközökre lenne szükség problémáik megfogalmazásához és megoldásához. Kielégíti-e őket egy jól megfogalmazott univerzális nyelv, vagy inkább speciális nyelvet érdemes kidolgozni a speciális problémákhoz? Hogyan lehet az ismétlődésektől, felesleges részletezéstől megszabadulni, és amit csak lehet, ráhagyni a gépre? Kidolgozható-e olyan eszköztár, amely a kész megoldást félkész elemekből rakja össze? Lehet-e az addigi gyötrelmes, hosszadalmas programírás helyett kész sablonokkal, géppel generáltatni a megoldóprogramokat?

Az USA-ban a magasszintű programnyelvek koncepciójának első sikerei nyomán a gazdasági és üzleti élet képviselői látták be elsőként, hogy micsoda fantázia van a programozás automatizálásában. (A számítógépeken végzett tevékenységeknek már akkor is több mint a fele az üzleti életet érintette.)

Boldog koboldok

1959-ben a Pentagon irányításával az adatfeldolgozás fejlődésének előmozdítására alakították meg a Codasyl-bizottságot, amely kilenc hónap alatt (!) szállította a közös üzletorientált nyelv, a Cobol (COmmon Business Oriented Language) első változatának teljes leírását. Számítástechnikailag a legnagyobb újítást a hierarchikusan felépített rekordstruktúra bevezetése jelentette. Nagy gondot fordítottak az adatállományok szervezésére és kezelésére, az adatbevitel és kiírás szabályozására, a számítási lehetőségek viszont lényegében a négy alapműveletre korlátozódtak. Az üzleti életben ugyanis rendszerint hatalmas adatmennyiséggel kell dolgozni, de csak viszonylag egyszerű műveleteket kell végrehajtani. Fontos szempontnak tekintették, hogy a feldolgozott anyagból a legkülönbözőbb csoportosításokban jól áttekinthető, könnyen értelmezhető eredménytáblázatokat lehessen előállítani.

A Cobol feltűnő szemléletbeli váltózást is hozott: a nyelv önmagát dokumentálta, s még a kívülálló is látta, hogy a program melyik része mit csinál. Az olvashatóság érdekében például minden tevékenységet az angolból vett igékkel (add, alter, close, compute stb.) fejeztek ki, ugyanakkor — az Algol körüli vitákból okulva — nagyon vigyáztak a Cobol felépítésének szerkezeti tisztaságára.

Nem sokat tétlenkedtek a gyártó cégek sem: elkészült az első fordítóprogram egy Univac gépre, majd sorra a többire. A fejlesztést az sem fékezte, hogy az eredeti nyelvdefiníciót a gyakorlati tapasztalatok alapján többször módosították. Lényegesebb változások azonban csak 1974-ben történtek, amikor az ANSI-szabványelőírásoknak megfelelően dolgozták át a leírást.

A Cobol fordítására kifejlesztett eszközöket szinte egy az egyben fel lehetett használni az Algol fordítóprogramjának kidolgozására is. Közben a formális nyelvek kutatói kiderítették, hogy a környezetfüggetlen szabályokkal felírható nyelveknek bizonyos feltételek teljesülése esetén mindig létezik olyan „jó viselkedésű" szabályrendszere, amellyel visszatérések nélkül elvégezhető a helyesen felírt „mondatok" szintaktikai elemzése.

Érdekes, hogy az amerikai számítástechnikus társadalom elméletellenessége a Cobol fordítóprogramok elterjedése nyomán szép lassan elpárolgott. Már a Cobol-ismertetésekben is rendszerint az Algolt, az Algollal való kapcsolatot emlegették. A Fortranról inkább tapintatosan hallgattak...

A Cobol is azok közé a „főnixnyelvek" közé tartozik, amelyek időről időre megújultak, először 1974-ben, majd 1985-ben. Így teremtették meg a különböző fájlszervezési módszerek közvetlen támogatását a Cobol programokból, az adatbáziskezelőkkel való kapcsolattartást SQL hozzáférési lehetőségek biztosításával. 1989-ben beépítettek a Cobolba jó három tucat függvényt, amellyel lehetővé tették a karaktermanipulációt, a dátumokon végzett műveleteket, valamint különböző üzleti és statisztikai elemzések elvégzését.

A közelmúlt és a közeljövő

Jelenleg a legjobb úton haladunk afelé, hogy az eredetileg tipikusan nagygépes Cobol nyelv szintje a PC-k számára is elérhető legyen. Bizonyára marad még tennivaló a nagygépek számára is, de az adatfeldolgozás javát ma már a kisebb kategóriájú gépek végzik el. Ismerünk olyan fejlesztést is, amely a régebben írt, hagyományos fájlkezeléssel működő Cobol programokat viszi át relációs adatbáziskezelő környezetbe (Cobol to SQL Transparency). Ezeknek a fejlesztéseknek központi problémája a Cobol nyelv köré épített megfelelő fejlesztői környezet kialakítása PC-ken vagy minigépeken.

Számos cég verseng, hogy melyikük tud kellemesebb környezetet biztosítani a fejlesztőknek. A MicroFocus nemrég jelentette meg 32 bites változatban új, objektumorientált Cobol-fordítóját. Az IBM-nél most készül egy Visual Cobol-szerű objektumorientált rendszer, a CA-nál pedig a Realia-Cobol stb. Hírek szerint 1996-ra lesznek készen az objektumorientált programozást támogató új Cobol-szabvánnyal. A jelek szerint az örökzöld Cobol élt, él és élni fog...

Egy kis kitérő

A szakmai specializálódás más területeken is létrehozott magasszintű nyelveket, amelyek azonban kisebb felhasz-

nálói kört érintettek. A specializáció helyett inkább vizsgáljuk meg a magasszintű nyelvek fejlesztésének ellenkező irányú tendenciáját, amely a potenciális felhasználói kör kiterjesztését tekintette feladatának.

Az látszott a legmegfelelőbb megoldásnak, ha magasszintű nyelvekből lehet minden problémát megoldani, amely a számítógépen egyáltalán megoldható. A magasszintű megközelítés előnyeit ezért kombinálni akarták az alacsonyszintű megközelítés előnyeivel (lehessen bitekkel manipulálni, ha a hatékonyság ezt kívánja).

Az alacsonyszintű nyelvekben valóban sokkal hatékonyabb programokat lehet írni. Például az Assemblyben minden lehetőséget ki lehet használni, amit a gépi utasítások megengednek, aminek azonban az az ára, hogy sokszorosára nő a program belövésére fordított idő. Ideális megoldásnak egy olyan kompromisszum megtalálása ígérkezett, ahol egyetlen magasszintű nyelvből elérhetők az alacsony szintű adatkezelési lehetőségek is, de olyan esetekben ne legyen a programozó ráutalva ezek használatára, ahol ez felesleges lenne. Az alacsonyszintű utasítások használata sem kell, hogy abszolút gépközeliséget jelentsen, a konkrét megvalósítás az adott gépen már a fordítóprogram dolga. (E probléma az operációs rendszerek írásakor vetődik fel élesen.)

Útkeresés

Térjünk vissza a hatvanas évekbe, amelyek páratlanul gazdagok voltak problémafelvetésben. A Fortran konszolidálása érdekében tett lépésekkel egyidejűleg az IBM sokkal nagyobb fába vágta fejszéjét: 1963-ban egy új, korszerű nyelv létrehozását határozták el az alábbi hármas célkitűzéssel.

a) Elégítse ki minél szélesebb programozói kör igényeit, beleértve a tudományos-műszaki és az üzleti-gazdasági feladatokat is.

b) A programírás nyelve legyen minél természetesebb.

c) Legyen alkalmas a jelenlegi (és a jövőbeli) számítógépek szoftvereinek, operációs rendszereinek megírására is. Így például lehessen megfogalmazni benne olyan operációs rendszer működését, amely valós időben végrehajtja az ugyanezen a nyelven megírt programokat.

Az ambiciózus célokban rejlik némi skizofrén vonás. Az első két cél mögött inkább üzleti meggondolásokat sejthetünk: olyan nyelvet képzeltek el, amely elhódíthatja a Cobol híveit, de megtartja a Fortran-programozókat is. Az igazi újdonság a harmadik pontban van, amely a nyelv erejének növelésére utal.

A monstrum

Erejét, „súlycsoportját" tekintve a PL/1 (ez lett az új nyelv neve) az Algol-68-cal vethető össze. Tulajdonképpen ez lett az első megvalósult, harmadik generációs nyelv. A mindenkinek eleget tenni akarás azonban magasabb szinten ismételte meg a Fortran eklektikusságát, a nyelvalkotók egységes koncepciójának hiányát. Mindenki megtalálhatja benne azt a zugot, amelyben a saját ízlése szerint programozhat, akár Cobol-, akár Fortran-stílusban. Ráadásul az egész nyelv ismerete nélkül, mert a nyelv különböző részei önmagukban is életképesek. A teljes adatelőírási szabadság azonban veszélyes is lehet. A számítógép erőforrásainak esztelen pazarlásához vezethet, és iszonyatosan rossz hatásfokú programot eredményez, ha valaki nem jól használja ki a nyelv lehetőségeit.

Hosszú ideig a PL/1 nyelvnek csak egy erősen beszűkített részhalmazát tudták implementálni. Igazi elterjedése nagygépeken is nehezen indult el, inkább csak akkor, amikor már rendelkezésre állt a Pliopt optimalizáló programrendszer. Operációs rendszer írásához pedig semmiképpen nem sikerült alkalmas nyelvet kifejleszteni belőle. A nagygépek háttérbe szorulásával pedig épp az a nyelv került szép lassan feledésbe, amelyet a Fortran és a Cobol felváltására hoztak létre. C'est la vie.

Hová tűnt el egy nyelv?

A PL/1 tervezői kezdettől fogva állandó munkakapcsolatban álltak egy angol csoporttal, amely jó egy évvel korábban kezdte el nyelvfejlesztő tevékenységét. Az angol csoport által tervezett nyelvről, a CPL-ről (Combined Programming Language) már 1963-ban megjelentek az első publikációk. Ekkor a PL/1-nek még halvány körvonalai is alig rajzolódtak ki néhány kutató agyában.

A két csoport tagjainak sok tekintetben hasonlóak voltak az elképzelései a létrehozandó nyelv tulajdonságairól, ezért rendszeresen találkoztak. Mindkét társaság nagyon hasznosnak ítélte a találkozókat, mivel azok lehetőséget nyújtottak a szemantikai finomságok megvitatására. Később azonban megszakadtak a kapcsolatok... A PL/1 megszületett, és sikert aratott, ha nem is kirobbanó sikert. Az angol programnyelvnek viszont mintha nyoma veszett volna.

1971-ben mégis felbukkant belőle valami a nyilvánosság előtt. Nem az egész CPL, csak a magja, a BCPL (Basic CPL), mint fordítóprogram írására alkalmas rendszerprogramozási nyelv. A furcsa csak az, hogy mindez már azután történt, hogy az AT&T Bell Laboratóriumában Ken Thompson felfedezte az eltűnt és halottá nyilvánított nyelvet, majd ennek alapján elkészítette a maga B nyelvét. A BCPL-ből tehát már a CPL is eltűnt. Viszont ezen a nyelven készült a PDP-7 első Unix operációs rendszere, amely alapja lett Dennis Ritchie C nyelvének. Nem különös? Mai korunk két legnépszerűbb nyelve, a Pascal és a C egyaránt európai ősökkel dicsekedhet.

Vargha Dénes

Számítástudomány — itthonról

A matematikusok és informatikusok számára a nyelvek formális leírása és egy alkalmas metanyelv kidolgozása jelentett komoly kihívást. Új matematikai diszciplínaként megszületett a formális nyelvek elmélete.

Idehaza Kalmár László vetette fel, hogy magukat a számítógépeket kellene úgy megkonstruálni, hogy kvázilogikai formulákkal vezérelhetők legyenek. A számítástechnika logikai megközelítésének másik hazai úttörője, Dömölki Bálint az első Magyarországon épített számítógép, az M-3 utasításkészletét egészítette ki két nagyon lényeges utasítással. Az egyik utasítás lehetővé tette, hogy bármelyik memóriarekeszt fel lehessen használni indexregiszterként, a másik pedig, hogy az indexregisztert is indexelni lehessen egy újabb memóriarekesszel.

Dömölki Révész Györggyel együtt elkészített egy szimbolikus címeket fordító programot, és kidolgozott egy elegáns univerzális szintaktikai elemző algoritmust, amely elsősorban logikai vektorokkal és stackekkel manipulált. Később ez az algoritmus lett az alapja több kisebb hazai fejlesztésű, szintaktikai kifejezésekkel vezérelt fordítóprogramnak.

Operációs rendszerek és fordítók

Az „igazi" programozók terepe

Azok a régi szép idők!
Amikor még az operációs rendszerek „komolyak" voltak, nem léteztek személyi számítógépek, csak óriási, több szobát elfoglaló, akkoriban nagy teljesítményűnek számító masinák, amelyek egy kisebb erőmű energiáját emésztették fel egy összeadás elvégzéséhez...
Azokhoz a gépekhez nagy-nagy operációs rendszereket adtak.
(Ha gyenge a hardver, a szoftver majd pótolja a hiányosságokat.)

A csöves számítógépek korában a programozás néhány ember kiváltsága volt. Ekkor írta Knuth „A programozás művészete" c. trilógiáját, és ebből az időből származik az a legenda is, miszerint Seymour Cray egy Honeywell gép mérnöki konzolján binárisan ütögette be a gép első operációs rendszerét.

Ilyen csöves számítógépeken, a nálunk fellelhető Ural-2-esen vagy Minszken már nem dolgoztam; az első számítógép, amelyhez hozzáfértem, egy lengyel gyártmányú Odra 1204-es tranzisztoros gép volt. Háttértárolója mágnesdob, az adatokat lyukszalagon lehetett „megetetni", az eredményt sornyomtatón kaptuk. A gépet sohasem láttuk, csak a hozzá tartozó operátort, akinek reggel leadtuk a „job"-ot, szerencsés esetben délután mehettünk a futtatási eredményekért. Időnként, ha sok futtatni való volt a gépen, akár napokat is kellett várni az eredményre. Akkoriban az „interaktivitást" nem a gép, hanem az operátor képviselte, akivel esetenként megbeszéltük a hibalistákat.

A gép operációs rendszeréhez „default"-ban tartozott egy-két fordító: az egyik az elmaradhatatlan Fortran nyelv fordítója volt (nem Fortran–77, hanem még csak Fortran–66), a másik a gép Assembly nyelvének fordítója (assembler). A gép nyelvét JAS-E-nek hívták, és lengyel gépről lévén szó, fordítója a lengyel mnemonikokat értette meg. Igen szörnyű volt, amikor az ember egy programban ilyeneket írt: WJAZ A,B; LEWJ; stb. Még szerencse, hogy a Műegyetemen valaki vette a fáradságot, és átírta angol mnemonikosra az assemblert. Aztán ott volt a Fortran, akkoriban más nem is létezett, hacsak nem az Algol. Fortranban persze elég nehéz volt nagyobb programokat írni, mert sehogyan sem akart beférni a számítógép memóriájába. Akkor az 1-2 Mbájt nem az operatív tár, hanem a mágnesdob kapacitását jelentette.

A PDP és az ESZR apropóján

Persze a komolyabb, nagyobb teljesítményű gépekhez is járt Fortran-fordító. Az első ilyen egy többfelhasználós gép volt, amelyen dolgozhattam: a Digital gyártmányú PDP-11/45. Paraméterei: teljes 256 kbájt ferritgyűrűs memória, 80 Mbájtos merevlemezes egység. (Nem winchester, hanem a régi CDC merevlemezes tár, amelyet „bolgár diszknek" hívtunk.)

Ehhez a géphez egyébként a Digital háromféle operációs rendszert is adott: az egyfelhasználós RT11-et, a többfelhasználós RSX-et (real-time változata már folyamatvezérlő számítógépekhez is létezett) és az RSTS-t (Resource Sharing, Time Sharing). Ez utóbbi egy olyan multitaszkos, multiuseres operációs rendszer, amelynek az alapnyelve a Basic volt, a gépen alapértelmezésben futott egy Basic-interpreter. Ez azt jelentette, hogy parancssor szinten lehetett Basic programokat írni, és nem is akármilyeneket.

Az operációs rendszerhez tartozott az elmaradhatatlan Fortran programozási nyelv is. Így már lehetett nagyobb programokat írni, mert az operációs rendszer gondoskodott a 256 kbájt memória mint „osztható erőforrás" megfelelő kiosztásáról, amelyet 64 kbájtos lapokra osztott. Mindenkinek ekkora lapokat osztott ki a rendszer, ha a time sharing úgy ütemezte, hogy az adott felhasználó programja futhat. Ezen a gépen nem volt ritka, hogy egyszerre 15-20 ember is dolgozott interaktív (!) terminálokon (VT52, VT100), és őket ki kellett szolgálni ésszerű időn belül. Ez azért követelt egyet s mást az operációs rendszertől.

A rendelkezésre álló erőforrást tisztességesen ki kellett használni, mert akkoriban a gépidő még elég drága volt. Egy mezei felhasználó előre foglalt gépidővel, naponta maximum két órát dolgozhatott a gépen. Akkoriban voltak még „igazi programozók", akik háromoldalas DO ciklust írtak Fortranban, anélkül, hogy belezavarodtak volna. Mellesleg az „Igazi Programozó" (Real Programmer) egy szövegfájl, amely az Interneten vagy a CopmuServe-en kering, magyarra fordított változata megtalálható minden magyar BBS-en.

A Fortran nem ismerte a dinamikus tárkiosztás fogalmát. Minden szubrutinnak előre, statikusan lefoglalt munkaterülete volt a memóriában, ha kellett, ha nem. Ebből az következik, hogy a Fortranban írt programok nem bántak éppen szűkösen a memóriával. A memóriakezelési feladatokat elsősorban az operációs rendszer látta el. Persze a komolyabb gépeken volt már MMU (Memory Management Unit) processzor is, vagyis hardverrásegítést kapott a rendszer. De elképzelhető, hogy mennyire takarékosan kellett bánni a memóriával (azzal az egyszerre rendelkezésre álló 64 kbájttal), amelyben még az operációs rendszer megfelelő darabjának is helyet kellett kapnia. (RSX-nél 8 kbájtban fért el a „command.com"-nak megfelelő rész, persze felhasználónként.) Az operációs rendszer feladata volt, hogy az éppen nem futó jobokat a diszkre ideiglenesen kitegye, majd ha ismét rákerült a sor, a memóriába behívja. Ugyanakkor minden jobot más-más prioritással, és más-más védelmi szinten (kernel, user) kellett kezelni. Mindezek ellenére sohasem sikerült lefagyasztani a PDP gépeket, legalábbis userszinten nem. Kernel (supervisor) szinten egyszer-kétszer azonban sike-

rült valamennyi gépen dolgozó felhasználó haragját kivívnom a gép lefagyasztásával. (De ez nem az operációs rendszer hibája volt... Akkor tanultam meg, mit jelent a rendszeres mentés.)

Szóljunk néhány szót az IBM-alapú ESZR gépekről is! Az IBM 360-as, 370-es család jóval nagyobb volt a PDP gépeknél, így az operációs rendszerek (OS/VS, POWER/VS és az első DOS) is nagyobbak és bonyolultabbak voltak. Ezekhez a gépekhez is járt assembler és Fortran, de ez a Fortran csak részben hasonlított a DEC-változatra. Ugyanúgy megvolt a maga „nyelvjárása", mint a mai programozási nyelveknek. Az IBM Assembly nyelvét pedig még az IBM gépeket programozók szerint is elég nehezen lehetett elsajátítani.

A 8 bites gépek korában

Később, amikor kezdtek elterjedni a kisebb, olcsóbb számítógépek, már az operációs rendszereket sem kellett olyan bonyolultra készíteni, mint például az R11-es gép MMTS operációs rendszerét, ahol egy egyszerű COPY parancs öt sor begépelését igényelte az operátori konzolról.

Az egyszerű gépek (az első mikroprocesszoros masinák: TAP34, MOD81, VT20 stb.) mind vagy Intel 8080-at, vagy ezzel felülről kompatibilis Zilog Z80-as CPU-t használtak. Operációs rendszerüket illetően azonban nem voltak ilyen egységesek, bár a nyugati szelek idefújták a Digital Research CP/M operációs rendszerét, amely gyakorlatilag szabvány lett a 8 bites architektúrájú gépekben.

A CP/M tényleg egyszerű operációs rendszer volt: egyfelhasználós, egytaszkos (kivéve a későbbi, v2.2 utáni változatokat). Azonban a gyártó alapértelmezésben nem adott hozzá semmiféle fordítóprogramot, azt mindenki onnan szerezte meg, ahonnan tudta. A CP/M minden platformon ugyanazt a tárkiosztást követelte meg, és ugyanazt a felületet adta, így könnyűvé tette a programok írását. A magasszintű nyelvet használók programjait CP/M-es gépek között viszonylag könnyen lehetett (forrásszinten) hordozni — feltéve, ha mind a forrásgépen, mind a célgépen rendelkezésre állt a megfelelő fordítóprogram (például a jól ismert Turbo Pascal).

A Unix és a C

A 8 bites gépek korában nem csak CP/M létezett. 1977-ben két úr, Thomson és Ritchie készített egy — eredetileg kis telefonközpontok vezérlésére szolgáló — operációs rendszert, amelynek a Unix nevet adták. Ehhez az operációs rendszerhez fejlesztette ki Kernighan és Ritchie a C programozási nyelvet. Ma mind a Unix, mind a C (illetve mai változata, a C++) a világ legelterjedtebb szoftvereszközei közé tartozik. Ami meglepő, hogy a Unix elsősorban nem a kisgépeknél hódít.

A Unix az az operációs rendszer, amely el sem képzelhető anélkül, hogy legalább egy C fordítóprogramot ne adnának hozzá. Persze a C nem volt mindig ilyen elterjedt, mint manapság. A Fortran nagyon sokáig tartotta magát, elsősorban a rengeteg Fortranban megírt alkalmazás miatt. Még a 80-as évek végén is olyan statisztikák láttak napvilágot, hogy a megírt szoftverek több mint 50%-a Fortranban íródott, beleértve az ún. beépített (rakéta- és műholdvezérlő) szoftvereket is. Pedig ekkortájt már a Pascal és a C igencsak a piacon volt, sőt a strukturált programozás őrületében a Pascal igen népszerű nyelv lett.

A C-t azonban nem a kezdőknek találták ki. A C-fordítók nem vezették kézenfogva a programozót, mint a Pascal nyelv fordítói. A C egyik megalkotója, Dennis M. Ritchie maga írta, hogy nem szeretett gépelni, és a lehető legegyszerűbb, legrövidebb kulcsszavakat választotta a nyelv leírásához. Statisztikát készített az egyes műveletek előfordulási gyakoriságáról: az értékadás operátor azért lett például egy egyenlőségjel (=), a feltételvizsgálat pedig azért lett kettő (==), mert ez utóbbi egy átlagos programban kevesebbszer fordul elő, mint az értékadás. Így a C persze nehezen olvasható. Talán épp az ellenkezője a CDL (Compiler Description Language) nyelvnek, amelyben a programozó kvázi a saját szavaival írja le a programot.

A C-fordítókból úgyszólván hiányzik a program helyességének vizsgálata, egyszerűen ellenőrzi a szintaxist, és szemantikusan helyes programot feltételezve kódot generál belőle (Aztec C). De hogy mi kerül a programba, a kész kód hol ír operációsrendszer-területre, az mind a programozóra van bízva. A C mindent megenged, mindent lehetővé tesz. Talán épp ezért nem Assemblyben, hanem C-ben írták a Unix több mint 90%-át. Ezért is forrt úgy össze a Unix a C-vel. Unix, illetve Unixszerű operációs rendszert manapság ezen operációs rendszerek gyártói (Unix, AIX, OSF, Ultrix, Solaris, Xenix, UNX, SCO Unix stb.) nem adnak C-fordító nélkül.

PC-s fejlesztői támogatás hiányában

Végül beszéljünk a mai PC-k legelterjedtebb operációs rendszeréről, a CP/M-ből származó DOS-ról, illetve a Macintosh operációs rendszerével rokonítható Windowsról. Nem tudom, kapott-e valaki az operációs rendszer mellé „bundle" fordítóprogramot, netán teljes fejlesztőkörnyezetet. A DOS korábbi verziói megelégedtek azzal, hogy az operációs rendszer mellé adták a link.exe task-builder programot, az exe2bin.exe konvertert, és a debug.exe hibakereső programot. És ezzel vége. De hogy mit linkeljen az ember, ahhoz már külön assemblert vagy magasszintű fordítót kellett vásárolni. A DOS 6-tól kezdve már exe2bin és link sincsen, nem beszélve a Windowsról, amely helyből semmilyen fejlesztői támogatást nem ad, még egy debuggert sem. Ehhez képest viszont a Windows önmagában is elég bonyolult, erőforrászabáló, sok hibával teli operációs rendszer. (Eltérően a Macintosh operációs rendszerétől, amely megjelenésében hasonló, sokkal jobban bánik az erőforrásokkal, és kezelése is egyszerűbb.) Nem tudom, ki hogyan van vele, de szívesebben vettem volna egy kis fejlesztői támogatást a Windowstól, mintsem hogy furmányos módon előcsalhassak belőle egy stáblistát.

Sík Zoltán

KÖVETKEZŐ SZÁMUNKBAN A HÓNAP TÉMÁJA:

CD-ROMTÁR

Meditáció a tudás jövőjéről

Programozás — alapok nélkül?

Klasszikus kérdés, hogy mire érdemes oktatni a nebulókat. Jól elvitatkoznak erről azok a tanárok, akik számítástechnikát tanítanak. Az egyik iskolába tartoznak a Basicből és más programnyelvekből vizsgáztatók. A másikba, akik inkább alkalmazási oldalról próbálnák közelebb vinni egymáshoz a gyerekeket és a gépeket. A vita közben a fejlesztések zavartalanul folynak, és a gépek egyre barátságosabbak lesznek az átlaggyerekhez, átlagemberhez. Lehetséges, hogy a vita magától elhal. No nem azért, mert az információs gyorsforgalmi út egyik oldaláról nem tudnak átbeszélni a másikra (mint ahogy az az autópályáknál tapasztalható), hanem mert elmosódik a különbség az alkalmazás és a programozás között.

A hirdetések, az újságcikkek egyre könnyebben és egyszerűbben kezelhető programokról, egyre okosabb fejlesztőrendszerekről szólnak. Érdemes elgondolkodni azon, hogy a következő években várható változások hogyan érintik a programozót. Lesz-e még programozóra szükség? Kiszorítják-e ezt a tevékenységet olyan elérhető árú eszközök, amelyekkel mindenki minden számítástechnikai problémáját meg tudja oldani? Azt a tevékenységet, amelyet programozásnak mondunk, el tudja-e végezni bárki, akinek van érettségije, de nincs a mai értelemben vett programozói felkészültsége, tudása?

A válaszom erre a kérdésre röviden: igen. Kicsit hosszabban: inkább nem. Elmélkedhetnék a Turing gépről, és arról, hogy végeredményben minden programozás, amit csinálunk. De nem teszem. Mindegyikhez kell egy keveset tudni, ismerni, sőt tanulni. Amit ember készít, azt nem tudjuk ösztönösen kezelni, irányítani, erre csak akkor vagyunk képesek, ha megtanuljuk a módját. Ennyiben tehát semmiképpen sem lehet tudás nélkül programozni. A tudás szintje azonban már vitatható, végiggondolható kérdés, mint ahogy a történetiség sem hagyható el a kérdés tárgyalásából.

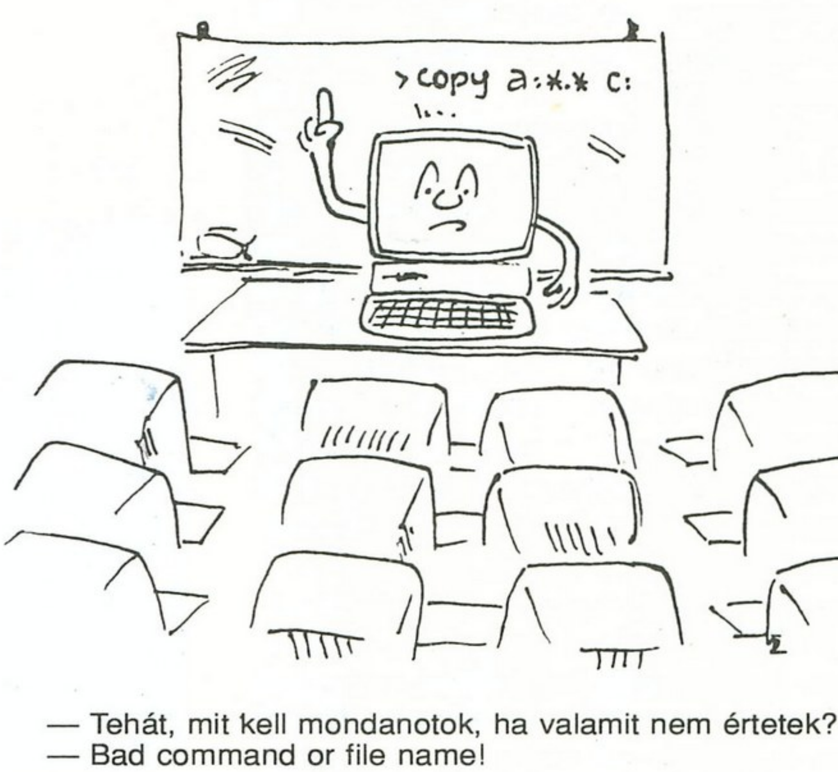
Programozás-e az, amikor egy teljesen grafikus kezelői felületen kijelölünk és kinyitunk egy mappát? Szerintem nem. Hiszen a programozás eredeti értelmében tárolt programozást jelent, vagyis azt, hogy a gépet nem közvetlenül vezéreljük, hanem a tárban elrakott utasítássorozattal. Mégpedig úgy, hogy a gép maga veszi elő, értelmezi és hajtja végre azokat. Programozásnak tekinthető-e egy számolótábla kezelése? Részben igen. Amikor egy rubrikába nem számot, hanem valamilyen függvényt, utasítást rakunk, akkor beprogramozzuk a táblázatkezelőt. Programozzuk-e a szövegszerkesztőt, amikor egy szót elhelyezünk a helyesírási szótárban? Bizony programozzuk.

A sor folytatható...

Akár a számítógépen kívüli — de igazából rokon — példákkal bőven élhetünk. Programozzuk-e a mosógépet, a tévékészüléket, a mikrohullámú sütőt vagy nem? De hiszen itt mindig egyetlen utasítást adunk, a gép azt hajtja végre, még akkor is, ha az, amit csinál, egy bonyolult műveletsor. Az egész bele van drótozva, tehát alaputasításnak számít. A tévékészülék előválasztóján már érdemes jobban elgondolkodni. Ez már tárolt programnak néz ki, de ha jobban megkaparjuk, akkor kiderül: ez sem az. Hiszen csak arról van szó, hogy előbb arra utasítom, miszerint egy rekeszbe tegye be valamelyik adás adatait, majd később, más időpontban parancsot adok neki: vegye onnan ki, és álljon rá az aktuális műsorra. Ha a televízió olyan, hogy napra, percre képes elraktározni, mikor mit mutasson, és mikor kapcsoljon ki, ennek beállítása már mindenképpen programozás.

Hiába csűröm-csavarom, nem lehet a világot úgy két részre osztani, hogy az egyikben vannak a programozni tudó emberek, a másikban pedig a többiek. Mégis, most az ezredvégen, megszabadulva a számítástechnika — mint a kiválasztott kevesek által ismert újdonság — körüli misztifikációtól, még mindig marad bennem egy érzés, hogy *mégsem mindegy, ki meddig „avatkozhat be" a dolgokba.*

Amit a szakma általában programozáson ért, az tanulás, gyakorlat, programozói tudás. Készség nélkül művelni nem lehet, és még sokáig lehetetlen is marad. A Webster New World Dictionary of Computer Terms (3. kiadás,



— Tehát, mit kell mondanotok, ha valamit nem értetek?
— Bad command or file name!

1988) szerint programozni annyit tesz, mint „lefordítani egy problémát a fizikai környezetből a számítógép által megértett és végrehajtható nyelvre. A probléma megoldásához vezető út megtervezése. Ez sok más között magában foglalja a probléma elemzését, kódolását, a be- és kiviteli formátumok, az ellenőrzés és kipróbálás megtervezését, a tárterület biztosítását, a dokumentáció elkészítését, és a kész program futásának felügyeletét." Ja, ha ez mind a programozás, akkor persze szakember kell hozzá.

És az objektumorientált programozás?

A már említett fejlesztőrendszerek, amelyek grafikus kezelői felületükkel, előregyártott elemeikkel a probléma megfogalmazásában, lefordításában is segítenek, szinte vezetik kezelőjüket. Vezetik, de hová? Természetesen a probléma megoldásához — szól a magától értetődő válasz. Valóban? Kétlem. Kételyeim az utóbbi időben nemhogy fogynának, de inkább nőnek.

Ami ugyanis így létrehozható, a probléma végiggondolása, kiérlelése nélkül, az csak véletlenül lesz az, amire valóban szükség van. Látványos példa a „ripsz-ropsz megvan" technológia veszélyére a rengeteg *chart*, amelybe lépten-nyomon belebotlik az ember. Lassan számoszlopokkal már nem is találkoznak az igazgatók, vezetők, csak diagramokkal. Csakhogy ezek értelmezése nem magától értetődő, sőt sokszor félrevezető is lehet.

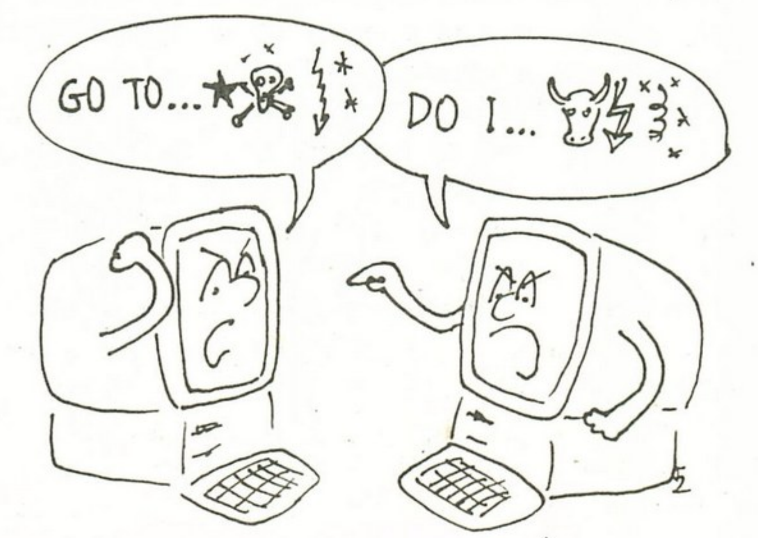
Jó-jó, fogadjuk el, hogy a probléma megfogalmazását, végiggondolását, a *rendszerszervezést* nem tudja mindenki elvégezni.

Na és a kódolás? Ha nincs a mai értelemben vett programírás, csak ide-oda húzogatása ennek-annak a képernyőn az egérrel? A programozásnak a kódolási részét, az utasítások leírogatását valóban felválthatja az új technika. De vajon az építőkockák, osztályok ellenőrzése, annak vizsgálata, hogy odaillenek-e pontosan arra a helyre, ahová szánjuk őket, nem visz-e el több időt, s főleg nem követel-e a kódoláshoz szükségesnél akár mélyebb, alaposabb tudást?

Most, hogy ezt a cikket írom, eszembe jutott egy aprócska malőr. Írtam valamit Arial nevezetű betűvel, amerikai Windows mellett, szövegszerkesztővel. A betűcsomag Arial CE (Central European), a billentyűzet Multikey-vel átdefiniálva, s lám *ő* betű helyett egy helyes kis négyzet, az *ű* betű helyett is egy helyes kis négyzet, nyomtatásban is. Újra telepítettem az Arial CE-t, ugyanaz. Betöltöttem a magyar Windowst, s megnéztem abban a szöveget. Megjavult...

Ahogyan a hibás Pentium processzornak lehetett lebegőpontos műveleti hibája, így a kész objektumokból létrehozott újjal is előfordulhat, hogy lyukra szalad, s rossz választ ad. Mi lesz azzal az objektummal, amely a kívánt választ adja akkor, amikor a laikus kipróbálja, csak éppen akkor csinál mást — mert például egyetlen apró környezeti paraméter megváltozott —, amikor a kész alkalmazást kell futtatni? És főleg mi lesz azzal, akit becsapott?

Amikor papírral és ceruzával számolunk, jobban bízhatunk az eredményben, mint amikor a számológéppel. De még az is megbízhatóbb, mint a számolótábla. A gépi utasítástól a Fortranig, az Adáig, C-ig kisebb az ugrás, mint ezektől az objektumokig, osztályokig, a tulajdonságok és a viselkedés öröklődéséig. Minél bonyolultabb egy feladat, annál félelmetesebb rábízni néhány fekete dobozra. Én bizony azt hiszem, jobb, ha vannak ott szakemberek, akikben meg lehet bízni, hogy elkerülik — és amennyiben nem, akkor legalábbis elviszik — a balhét!



Nyelvek és programnyelvek bábele

Nyelvében él a nemzet, mondogatják azok a nyelvújítók, akik a számítógépes programok magyarításáért kardoskodnak. Ugyanakkor — milyen érdekes — nem teszik ugyanezt a programnyelvek parancsainak nemzetiesítésével. Még a gyermekeknek tanított nyelv, a PC Logo is angol kifejezéseket alkalmaz.

A programozási nyelvek alapnyelve az angol. Elindultunk az Assemblytől, amely számokat és angol rövidítéseket tartalmazott, és elérkeztünk olyan parancsnyelvekhez, mint például a Lotus HAL interfész, amely a beszélt amerikai angol nyelv mondataiból ért.

Amikor a kínnal-keservvel magyarított (vagy máskor és máshol a németesített) programokat látom, felemás érzéseim támadnak. Az applikációk egy része beszél csak nemzeti nyelven. Viszont a keretrendszerek nemzetiesítése szinte gettóba zárja a számítógép-alkalmazót: mintha például valaki azt követelné, hogy x városban csak y nyelven lehessen beszélni.

A számítástechnika programnyelvei a természetes nyelvek felé haladnak. Ugyanakkor előttünk áll a rendhagyó igék, szókapcsolatok problémája, minden nyelvet tanulónak feladva a leckét. Ezekben a mi nyelvünk is gazdag, de van furcsaságok tárháza az irodalmi angolban éppúgy, mint a németben vagy a franciában. És ilyenkor mit lehet tenni? Megtanítani rá a gépet. Ha gépünk már eléggé okos, akkor esetleg már nincs memóriája arra, hogy valamit csináljunk is vele...

A probléma megoldását többen a műnyelvekben, így az eszperantóban látják, ahol a rendhagyó esetek száma nulla, és minden jól algoritmizálható. Nem véletlen, hogy a Közös Piac-i automatikus szövegfordító rendszerek az eszperantót használják belső nyelvként. Próbálkozhattak volna az újlatinnal is, de annak nyelvtani szerkezete kemény dió mindenkinek.

Milyen nyelven beszélnek ezután majd a gépek? Nyilván, ha programozni akarjuk őket, akkor angolul. Nemcsak hagyománytiszteletből, hanem ésszerűségből is. Ugyanakkor a programok forgalmazási logikája ez ellen dolgozik. Minél több egymással inkompatibilis változat van a piacon, annál többet lehet belőle eladni. Hát akkor: mi győz? A logika vagy az értelem?

Kis János

Osztály, öröklődés, többrétegűség

Az objektumorientáltság genetikája

A mindenkori programozók is beleütköznek abba a teljesen általános, az élet minden területén felbukkanó problémába, hogy többet, nagyobbat, jobbat akarnak alkotni, mint amire addig alkalmazott módszereikkel és eszközeikkel képesek (lehetnének).

A megoldási alternatívák sokfélék. Voltak, akik a hangsúlyt a módszertanra helyezték. Ebből a megfontolásból alakult ki például a strukturált programozás elmélete, amely sok módszerbe és módszertanba beépülve most is jelen van. Mások a feladathoz fejlesztettek eszközöket, a programozást próbálták kiiktatni a számítógépes problémák megoldásából — és néha komoly (rész)eredményeket is felmutattak.

Ezekkel a törekvésekkel párhuzamosan és együttélve fejlődtek a programozási nyelvek. Próbálkoztak a nyelvek vezérlési struktúrájának alapvető megváltoztatásával (Prolog, Forth), de nagyobb eséllyel azok a megoldások szállhattak ringbe, amelyek a már bevált, megszokott programozási nyelveket kiterjesztve adtak lehetőséget a programozóknak új módszerek alkalmazására. Ezek közül a legsikeresebbnek az objektumorientált programozás bizonyult, amely sok különböző megvalósítási mód ellenére néhány lényegi elemében azonos.

Az objektumorientált programozás tehát elsősorban módszertan, de létezése elválaszthatatlan azoktól a nyelvektől, amelyek támogatják. Alapgondolata, hogy a programon belüli adatstruktúráknak jobban kell támogatniuk a való világ modellezését, mint ahogy azt a programozási nyelvek jó része lehetővé teszi. A módszer sikerét mi sem jellemzi jobban, mint az, hogy ma már a C-fordítót kínáló cégek egyike sem érzi piacképesnek termékét objektumorientált kiterjesztés (C++) nélkül.

A tárgy mint „munkadarab"

A programokon belül rendszeresen beleütközünk abba a problémába, hogy a való világot kell valamilyen egyszerűsített módon modellezni. A való világot lássuk mondjuk tárgyakból álló valóságnak. Legyenek a programban is tárgyak, dolgozzunk azokkal. Na jó, de mi az, hogy tárgy? Mivel lehet egy tárgyat megadni? A legegyszerűbben a tulajdonságaival (szín, súly, szag stb.), és azokkal a műveletekkel, amelyeket a tárgyon el lehet végezni. Ha ezt mind felsoroljuk, akkor tökéletesen leírjuk a tárgyat.

Ugyanezt kell tenni egy objektumorientált programban is. Azt a valamit, amit le akarunk írni, tulajdonságainak és a rajta végezhető műveletek összességeként kell leírnunk. Lényeges, hogy a művelet (és leírása) valóban része a tárgynak, attól el nem választható. Az, hogy valamit meg lehet enni, éppen olyan lényeges információ lehet, mint az, hogy miként lehet ehetően elkészíteni. Ez a két dolog a programozási nyelvekben általában úgy jelenik meg, hogy az objektum leírása formailag struktúradeklarációra emlékeztet, de benne változók és függvények egyaránt szerepelnek. Ezeket a függvényeket szokták módszernek (method) is hívni.

A sokaság és az egyed

Az objektumok általánosítását, azaz típusát osztálynak hívják, egy osztályból létrehozott objektumot pedig az adott osztály egy példányának is szokták nevezni. A legegyszerűbb osztályok általában az alaptípusok: egész szám, karakter, karakterfüzér. Egy kicsit már bonyolultabb osztálynak tekinthetjük a komplex számokat, ahol a valós és a képzetes rész az a két változó, amely az objektum tulajdonságait alkotja, illetve a valós számok összeadására, szorzására stb. vonatkozó szabályok alkotják a rajtuk elvégezhető műveletek halmazát, azaz módszereket, illetve C++-elnevezés szerint tagfüggvényeket.

Az osztály lényegesen több, mint egy struktúra-típusdeklaráció, mert modulként is funkcionál. Ennek megnyilvánulása, hogy például a C++-ban minden osztálynak szabályos felülete van. Azaz meg lehet szabni, hogy melyek azok a változók és függvények, amelyek csak az adott osztályból származó objektumokon belül használhatók (ezek az ún. private függvények és változók), illetve melyek azok, amelyek mindenki által használhatók, és melyek a csak a származtatott osztályokban használhatók (a protected függvények és változók). Így, ha valaki objektumorientált programot akar írni, nem modulokban, hanem objektumokban kénytelen gondolkodni.

Származások, leszármaztatások...

Ha valaki — mondjuk egy ablakkezelő — programozva készít alkalmazást, akkor a programjában feltehetőleg több olyan valami lesz, amit gyűjtőnéven csak „ablak"-nak szoktunk nevezni. Ennek egy nagyon leegyszerűsített változatát C++-szerű formában valahogy így lehetne leírni:

```
Osztály Ablak {
  Pozíció poz;
  Dimenzió dim;
  Mozgatás( Pozíció poz );
  Méretezés( Dimenzió dim );
}
```

ahol a Pozíció és Dimenzió szintén egy-egy osztály, deklarációjuk az alábbi lehet:

```
Osztály Pozíció {
  szám x;
  szám y;
}
Osztály Dimenzió {
  integer szélesség
  szám magasság;
}
```

Abból, hogy egy tárgy leírásához annak minden lényeges tulajdonságát és módszerét fel kell sorolni, sok kényelmetlenség is adódhat. Gondoljunk arra, ha például le akarjuk írni egy épületen belül az összes ajtót, akkor gyakorlatilag

minden különböző típusú ajtóra új osztályt kell deklarálni. Holott valamennyi ajtó rendelkezik olyan közös tulajdonságokkal, amelyeknek csak értéke különbözik: például minden ajtónak van színe, magassága, zárja stb. Ráadásul bajba kerülhetünk, ha valaki egyik napról a másikra kitalálja, hogy minden belsőtéri ajtónak fehérnek kell lennie, mert akkor minden ajtótípust végig kell bogarászni.

Erre a problémára jó megoldás az öröklődés. Ennek lényege, hogy nem hozunk létre különböző típust minden ajtóra, hanem készítünk egy általános ajtóosztályt. Ez minden olyan általunk lényegesnek tartott tulajdonsággal rendelkezik, amely az épületben található bármelyik ajtót jellemezheti: van színe, magassága, szélessége, jobbról vagy balról nyílik stb. Ezt az ajtóosztályt további ajtóosztály megalkotására használjuk, és az így kapott osztályokat felhasználhatjuk újabb osztályok deklarálására. Ezt a lehetőséget hívják úgy, hogy öröklődés, míg az öröklődéssel létrehozott osztályokat származtatott osztályoknak nevezik.

Az öröklődés nagyon lényeges eszköz. Ha jól választjuk meg azokat a jellemzőket, amelyeket beleveszünk ős- vagy alaposztályainkba, akkor nagyon sok munkát megspórolhatunk. A származtatott osztályok használhatók minden olyan helyen, paraméterpozícióban, ahol ősüket kellene használni, de speciális saját tulajdonságaik ezeken a helyeken természetesen nem használhatók ki.

Vissza a példára!

Nyilvánvaló, hogy fenti Ablak osztályunk erősen hiányos. De ez nem baj, hiszen a deklarációhoz ráérünk valamit hozzátenni később is, más olyan osztályokban, amelyeket az adott osztályból származtathatunk, például a következő módon:

```
Osztály GrafikusAblak : Ablak {
    Rajzolás();
}
```

A kettőspont és a nyitó zárójel között van az az osztály, amelyik őse az új osztálynak. Így egy olyan osztályhoz jutottunk, amely mindent tud, amit egy Ablak osztály tud, példányai mindenhol használhatók, ahol ősének, az Ablak osztálynak a példányait kellene használni. És még valamit tud, lehet bele rajzolni, ha megírjuk hozzá a függvényt.

Másik példa

Használjuk ki azt, hogy a pont egy olyan valami, amelynek a matematikusok szerint nincs mérete, a programozók szerint viszont 1 képernyőpont méretű. Tehát mindenképpen adott, és van színe. Ebben az egy dologban különbözik a Pozíciótól, tehát azt színnel kiegészítve az alábbi Pont osztályhoz juthatunk:

```
Osztály Pont : Pozíció {
szám: szín;
Rajzolás();
}
```

Ha egy osztálynak nem csak egy osztály az őse (amelyet nem minden objektumorientált nyelv enged meg), akkor többszörös öröklődésről beszélünk. Ilyenkor a származtatott osztály mindkét ősének tulajdonságait örökli. Fenti osztályainkat például definiálhattuk volna az alábbi segédosztállyal így is:

```
Osztály: Rajzolható {
    Rajzolás();
}
Osztály GrafikusAblak : Ablak, Rajzolható {
}
Osztály Pont : Dimenzió, Rajzolható {
}
```

Ennek természetesen főleg akkor van értelme, ha a Rajzolható osztályt más osztályok definiálásakor fel akarjuk használni.

Többrétűség

Az osztályhierarchia tervezésekor az általános osztályoktól a konkrét osztályok felé haladva egyre több részletet tisztázunk. Ez azt jelenti: bizonyos dolgokról már egy általános célú objektumnál is tudjuk, hogy „kell egy olyan függvény", és azt is tudjuk, hogy mikor akarjuk használni, de azt még nem, miként kell megcsinálni. Előfordulhat, hogy valamelyik függvényt a származtatott osztályban át kell definiálni, hogy például kezelni tudja az újonnan definiált változókat. Erre a problémára több megoldás is létezik, amely mind ugyanazt az eredményt ad(hat)a, csak egy kicsit különböző módon.

Az egyik megoldás: a függvény deklarációjánál meg kell adni, hogy ha a származtatott osztályokban is van azonos nevű függvény (C++-ban még azonos típusúak a paraméterei is), akkor a két vagy több függvény közül a konkrétabb hívódik meg végrehajtáskor. Ezeket a függvényeket szokták virtuális függvényeknek is hívni. A másik megoldás szerint elég csak a hívásnál megadni azt, hogy a megadott osztálytípusból származtatott osztályban definiált függvényeket kell ténylegesen végrehajtani. Ezt szokták kései függvényválasztásnak, angolul *late bound call*-nak hívni. Van olyan megvalósítás, ahol a származtatott osztályban definiált függvények automatikusan átdefiniálják az őseikben létrehozott függvényeket.

Visszatérve korábbi példáinkhoz, a RajzolhatóAblak osztályban jó eséllyel nem tudjuk még megmondani, hogy mit és hogyan akarunk rajzolni, de azt tudjuk, hogy valamit rajzolni akarunk. Ezért vezettünk be egy rajzolás függvényt. Továbbra is a C++-szerű jelölésnél maradva, egy kicsit módosítsuk osztályunkat:

```
Osztály GrafikusAblak : Ablak {
    virtuális Rajzolás();
}
```

A szemléletesség kedvéért definiáljunk még két osztályt:

```
Osztály Alaprajz : GrafikusAblak {
    virtuális Rajzolás();
}
Osztály Térkép : GrafikusAblak {
    virtuális : Rajzolás();
}
```

A következőt értük így el: most már készíthetünk egy olyan függvényt, amelynek paramétere GrafikusAblak, és az ablakot újrarajzolja (és csinál még valamit), anélkül, hogy tudnia kellene, hogy az az ablak nem GrafikusAblak, hanem Alaprajz vagy Térkép, vagy valami más.

Speciális függvények

Az objektumorientált nyelvekben — sajnos nem mindegyikben — vannak speciális függvények, amelyek automatikusan meghívódnak az osztály egy példányának létrehozásakor vagy törlésekor. Ez a konstruktor és a destruktor. Ezt a két függvényt lehet arra használni, hogy az objektum belső változóinak kezdőértéket adjunk, vagy törléskor felszabadítsuk a elfoglalt tárterületet, letakarítsuk a képernyőt.

Az egyes objektumorientált nyelvekbe a fentieken kívül belekerült még sok egyéb sajátosság, de a lényeges, közös mag közöttük az osztály, az öröklődés és a többrétegűség fogalma.

Kubovics András

Az Awk programnyelv

Hatékonyságra törekvőknek

Hosszú ideig a sorszerkezetű szöveges fájlok voltak egyeduralkodóak a számítástechnikában. A saját fájlformátummal megjelenő sokféle felhasználói program sokat változtatott ezen, de azok is képesek fogadni és kibocsátani ASCII fájlokat is. (Itt most nem a grafikus programokra, hanem a szövegszerkesztőkre, az adatbáziskezelőkre, a számolótáblákra gondolunk.)
Ráadásul ezzel a fájlformátummal lehet leginkább átvinni az adatokat a különböző operációs rendszerek között, az utóbbi időben pedig a nagy nemzetközi hálózatok használatának terjedésével tovább nőtt a jelentősége:
a WWW (World Wide Web = világméretű „pókháló") dokumentumai is ilyenek, kizárólag karakteres információkat tartalmaznak.
Az ember nem ír compilert Clipperben (bár éppen lehetne), vagy nem kezd adatbáziskezelő készítésébe Assemblyben (bár lehetne).
A kimondottan szöveges, sorszerkezetű fájlok manipulálására is meg kell találni a megfelelő eszközt. Erre a célra született az Awk.

Az Awk nevet a program három „atyja" tiszteletére nevük kezdőbetűjéből képezték: Aho az elméleti számítástudomány egyik nagyja, Weinberger és főleg Kernighan a C és a Unix születésénél bábáskodott. Az Awk nyelv 1977-ben eredetileg azért készült, hogy a grep és a sed nevű Unix-beli eszközöket általánosítsák. Használhatósága azonban messze túlnőtt ezekén, egyre több helyen alkalmazták. Így 1985-ben újabb verziója készült el, amelyet egyes Unix gépeken nawknak (new Awk) hívnak, ma már ez a szabvány. A nyelv minden létező platformon megtalálható, van DOS-ra is jó néhány implementációja.

Közülük említést érdemel a Gawk, amely szabadon hozzáférhető, és az eredetileg a Polytron által kifejlesztett változat, amelyet azután a Thompson Automation vett meg és fejlesztett tovább. Ez kereskedelmi szoftver, fordítóval, debugerrel, mindennel együtt 200 dollár körüli áron megvehető.

A nyelv karaktere

Mit kell egy szövegfájllal elsősorban csinálni? Természetesen sorról sorra végigolvasni. Az Awk pontosan ezt teszi. Nem kell a fájlokat megnyitni, lezárni, külön olvasni, fájl végére figyelni. Ez mind automatikus. Egy Awk program szerkezete elég egyszerű. Vannak „minták", és ezekhez „tevékenységek" tartozhatnak. A program végigolvassa a parancssorban megadott fájlokat, mindegyiknek minden sorát minden egyes megadott mintával összehasonlítja, és amelyik sor illeszkedik egy mintára, arra végrehajtja az ahhoz tartozó műveletet. Lássunk egy egyszerű példát!

```
/^[A-Z]/  { nnagybkezd++ }
/[0-9]/   { ++vanszam }
END       {
              print "Nem nagybetűvel kezdődik",nnagybkezd," sor"
              print "Számjegy van",vanszam," sorban"
          }
```

A példa egy teljes program! Ebből rögtön kiolvashatunk néhány dolgot. Egyrészt az Awk szintaxisa hasonlít a C-re, így jó előiskola a C-hez, illetve egy C-programozónak semmi fáradságába nem kerül megtanulni az Awkot. Másrészt — és ez az Awk egyik legfontosabb erénye — a minták reguláris kifejezések is lehetnek. Ezek ismertetésétől most eltekintünk, de megjegyzendő, hogy például azt, hogy egy sztring egy lebegőpontos szám — esetleg exponenssel együtt felírt — alakja-e vagy sem, egy rövid reguláris kifejezés eldönti. Van két speciális minta, ezek a begin és az end. A beginhez tartozó tevékenységek egyszer, minden egyéb olvasás előtt hajtódnak végre (ideális változók inicializálására, felhasználói input kezelésére). Az END-re az összes olvasás után kerül sor, itt lehet például összegzések eredményét megjeleníteni. A változókat nem kell deklarálni, és típusuk sincs. Menet közben jönnek létre, és az értékadás során kapnak típust, ami bármikor változhat. Mint látható, van post és prefix inkrementálás is. Ez tömör, de még jól olvasható kódot eredményez.

Az Awkban számos beépített változó van. Az aktuális sor értékét a $0 változó tartalmazza, a szavak pedig sorban $1, $2 stb. A szavak számát az NF (number of fields) adja meg. Az éppen feldolgozott fájl neve a filename, a rekord sorszáma pedig az fnr változó tartalma. (Az Awk érzékeny a kis- és nagybetű különbségére, ezzel vigyázni kell!) Az előbbiekből látszik, hogy egy beolvasott sor rögtön szavakra bomlik.

Mit tegyünk egy olyan inputtal, ahol a mezőket nem szóköz választja el, hanem mondjuk a „ * " karakter. A szerzők erre is gondoltak. Az FS (field separator) változónak mi adhatunk értéket (menet közben is), amely szintén lehet reguláris kifejezés, így a szavakra tördelés módja teljesen kézben tartható. A következő példa egy egyszerű statisztikai program. Az input sorai ilyen alakúak, a mezőket tabulátor karakter választja el:

```
iroda dolgozok_szama bertomeg
```

A program megadja a cégnél a dolgozók átlagos fizetését.

```
BEGIN     {  FS="\t"
          {    dsum+=$2; bsum+=$3 }
END       {
               printf("Átlagkereset a cégnél: %8.2f\n",bsum/dsum)
          }
```

Mint látható, az output formázható is, a printf számtalan formázási lehetőséggel bír. Ha egy tevékenység előtt nincs minta, akkor az minden sorra végrehajtódik. A fenti program nemcsak rövid, de egy rutinos programozó fél perc alatt megírja, és utána nyugodtan eldobhatja, a következő elkészítése sem tart tovább.

A ciklusszervezés apropóján

A következő, összetettebb példa a nyelv szinte minden erősségét felvonultatja. Van egy szövegünk, amelyből szeretnénk szóstatisztikát készíteni: melyik szó hányszor fordult elő, és a szavakat ábécésorrendben kívánjuk látni. Mielőtt megnézik a programot, gondolkozzanak el, hogy kedvenc programnyelvükön mennyi idő alatt, milyen hosszú programmal oldanák meg a feladatot! (A szöveget tartalmazó fájl akármilyen nagy is lehet, a szereplő szavak számát nem ismerjük előre.)

```
          {
                 for (i=1; i<=NF; i++)
                       szavak[$i]++
          }
END       {
                 for (i in szo)
                       print i,szavak[i]
          }
```

Mint látható, az Awkban is van ciklus. A $i változó az i. szó. A „szavak" egy tömb, amit egyébként nem kellett deklarálni. De mik ennek a tömbnek az indexei? Szavak, természetesen. De hiszen eddigi tapasztalataink szerint egy tömb indexei csak egész számok lehetnek. Az Awkban ez nem így van! Az ún. asszociatív tömb bármivel indexelhető.

Ha még nem volt olyan szó, akkor létrejön a tömbelem, értéke pedig 0, illetve az üres sztring lesz, ez minden változóval is így van. Ha már volt ilyen elem, értéke eggyel megnő. Érdekes az END rész is. Ha egy asszociatív tömb minden elemére szükség lenne, akkor nem tudnánk, hogyan is indexeljünk. Ezért az (x in tomb) kifejezés sorban az indexértékeket adja az x változónak, azzal pedig már tudunk mit kezdeni. A tömb mérete nem korlátozott, a legtöbb Awk-változat még a lemezes swap fájlt is létrehozza, ha arra van szükség. Ha egy tömb már nem kell, akkor a delete tomb paranccsal egyszerűen megszabadulhatunk tőle. Lehetnek többdimenziós tömbök is (legalábbis a legtöbb Awk-implementációban), ezeknek az elemeit a

```
for (i in tomb)
     for (j in tomb[i])
          {
          ....tomb[i][j]
          }
```

szerkezettel lehet elérni. Így könnyedén lehet például kereszttáblát készíteni. A sorok első szava mondjuk egy megyekód, a második szava egy foglalkozás kódja (vagy akár neve). Azt, hogy megyénként hogyan oszlanak meg a foglalkozások, a következő egyszerű programrészlettel lehet kiszámítani:

```
          {
               m[$1][$2]++
          }
END       {
               for (i in m)
                    for (j in m[i])
                         print i,i,m[i][j]
          }
```

(Itt persze a formázás sor és az oszlopösszegek kiszámítását mellőztük.)

A for ciklusnak sokféle alakja lehet, hasonlóan a C-ben megszokotthoz. Természetesen van while és do ciklus is. A break és a continue utasítások a ciklusból való kilépést, illetve a ciklusmagból való kilépést vezérlik. Az if az elvárt módon működik. Fontos még a next utasítás, hatására a következő sort olvassa a program, a további mintákat és tevékenységeket átugorva.

Mi a teendő, ha szöveges fájlunk rekordjai többsorosak, és a rekordokat egyetlen üres sor választja el? Ráadásul különböző számú sor tartozik egy rekordhoz. Például egy könyvtári adatbázis sorai valamilyen azonosítóval kezdődnek (SZ mint szerző, CI mint cím, stb.), de egy-egy rekord nem feltétlenül az összes lehetséges azonosítót tartalmazza, és persze a sorrend is esetleges.

Szeretnénk egy listát az összes angol nyelvű könyvről (NY=angol), a szerzők ábécésorrendjében. Ezt a feladatot is mintha az Awkra találták volna ki. Az RS belső változó adja meg, hogy mi választja el a rekordokat. (Ez alapértelmezésben a "\n", azaz a newline karaktersorozat, ha kocsivissza és soremelés is van, mint a DOS-ban.) De ha ez az üres sor kell legyen, akkor az RS="" értékadás után az olvasás már mást tekint egy rekordnak, éppen azt, amelyik nekünk kell. Persze ilyenkor a mezőszeparátort is állítani kell. FS="\n". Ezek után a könyvtári adatbázis egyes sorai már mezőként vannak bent. A teljes program ismertetését elhagyom, de egyébként nem több 15 sornál.

Egy kis függvénytan

Az Awkkal való munkát számos belső függvény segíti. Ezek közül azok a fontosak, amelyek sztringek manipulációjára szolgálnak. A leggyakrabban használt függvény a substr, amely egy sztring megadott részét adja vissza. Hasznos a split, amely egy sztringet elemeire tördel, és az elemeket egy tömbben helyezi el. Persze azt, hogy mi szerint kell tördelni, szintén reguláris kifejezés adja meg. Így lehet egy algebrai kifejezést kényelmesen elemeire bontani. A sub, illetve gsub függvények sztringek részeinek átírását, helyettesítését végzik. A gsub az összes előfordulást, míg a sub csak az elsőt módosítja.

Az új, ma már standard Awkban a felhasználó is definiálhat függvényeket, s a függvényeknek paramétereik is lehetnek. Egyébként e definiált függvények — a C-nél megismert voidnak megfelelően — nem is kell, hogy értéket adjanak vissza, azaz eljárásként is funkcionálnak. Ez nagymértékben segíti a programok olvashatóságát, még inkább fejleszthetőségét.

Az Awk ideális a prototípusok elkészítésére, mert nagyon gyorsan lehet programot írni és tesztelni. Ha már jó a lényeg, a program majdnem egy az egyben tehető át C-be, és ott tovább tupírozható. Szintén ideális egyszeri, hamar megoldandó adatmanipulációkra, leválogatásokra, összegzésekre. Mivel maximálisan támogatja az input és az output átirányítását — ahogy ezt egy Unix-eredetű szoftvertől el is várjuk —, ezért prímán működik együtt más, mondjuk, rendszerprogramokkal. A nyelv ideális kompromisszum: gazdag eszköztár és egyszerű használat briliáns egyensúlya jellemzi. Érdemes megismerni, és ha éppen kell, használni! (Mostani lemezmellékletünket éppen ezért egy shareware-Awk is gazdagítja...)

Horlai János

Az Oracle CASE fejlődése

Vizsgálati szempontok (döntéskor is)

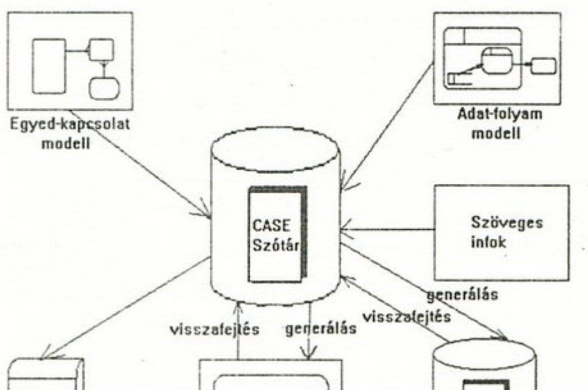
1992-ben egy cikkben megkockáztattam a következő kijelentést: „Tavaly, ha nagyon lassan is, megmozdult valami a hazai CASE-piacon." Három év elteltével sem ragadtathatjuk magunkat a fenti kijelentésnél sokkal optimistább megjegyzésre... Az eladások száma tényleg növekszik, a köztudatba bevonult a CASE fogalma. Nagy tenderek, új fejlesztések esetén általában megvásárolják, de sok esetben csak divatból, és nem komoly szakmai megfontolásokból teszik ezt. Éppen ezért most segíteni szeretnénk a CASE-vásárlást fontolgatóknak: egyrészt ismertetjük az alapvető fogalmakat, másrészt egy konkrét CASE-eszköz fejlődése kapcsán bemutatjuk a mai színvonalat.

Definiáljuk először a CASE (Computer Aided Systems Engineering) betűszó jelentését! A CASE jelenti a szoftverkészítési folyamat fázisokra bontását, a fázisokhoz különböző technikák használatát, jelent dokumentációs szabványokat és a minőségbiztosítást, valamint a fentiekhez számítógépes támogatást, prototípus-készítést és kódgenerálást.

Sajnos ma Magyarországon, de szerte a világban előszeretettel használják a CASE-eszközöket pusztán a második komponens értelmében, és elfeledkeznek arról, hogy egy ilyen eszköz a kemény módszertani megalapozottság és fejlesztői eltökéltség nélkül nem több, mint divatos játékszer. Bízva azonban abban, hogy egyre több megalapozott fejlesztés indul nálunk is, próbáljuk az alábbi elemzéseket megtenni.

CASE-környezet

A vizsgálati szempontok felállítása előtt a CASE-környezetek általános felépítését mutatjuk be az alábbi ábrán.



Nézzük meg, hogy a CASE-eszközök milyen komponensekkel rendelkeznek, és azokat milyen szempontok szerint érdemes vizsgálni!

A rendszer lelke a CASE-szótár, amely tartalmazza a rendszerelemzés és -tervezés során előállított és bevitt információkat, legyenek azok későbbi kódgenerálás kiinduló adatai, vagy „csupán" elemzési információk. A szorosan vett szótár vizsgálatánál két fontos szempontot érdemes figyelni. 1. A szótár tartalmi felépítését, azaz hogy milyen objektumokat milyen adatkörrel lehet benne tárolni. 2. A szótár megvalósítási módját. (Saját fájlrendszerrel vagy valamelyik elterjedt adatbáziskezelővel; publikus módon, azaz felhasználók számára is definiált szerkezettel van-e megvalósítva.)

A CASE-szótárba az információk bevitele kétféle módon történik. Vagy grafikus editorokkal, amelyek a CASE-eszközzel támogatott módszertan(ok) grafikus elemzési és tervezési technikáinak használatát segítik elő, vagy hagyományos képernyős bevitellel (ún. szótár interfésszel) további információk megadására.

A CASE-szótárból nyert outputoknak három fajtája ismert: rendszerdokumentáció (illetve minőség-ellenőrzési riport), generált adatbázis, generált alkalmazás.

A CASE-eszközök egy része rendelkezik egy speciális, fordított irányú kapcsolattal is a meglevő rendszerekkel. Nevezetesen, kész adatbázisokat és alkalmazásokat lehet segítségével a CASE-szótárba „visszafejteni", újratervezni, vagy csak egyszerűen beépíteni a készülő alkalmazásokba.

Az első verzió

A fentiek alapján a következő szempontokat vizsgáljuk az Oracle CASE fejlődése során: *szótárat* (tartalmat, user interface-t, a megvalósítás módját), *grafikus editorokat* (milyen technikákat támogatnak, milyen tulajdonságokkal rendelkeznek), *generátorokat* (milyen adatbázissémát generálnak, milyen 4GL-re generálnak kódot). Foglalkozunk a kiemelt platformokkal, amelyeken egy-egy verzió működik, de szeretnénk ismertetni az(oka)t a változás(oka)t is, amely(ek) egy új verzió elkészítését kiváltották.

Magyarországon 1989-ban a 4-es verziójú Oracle CASE-eszköz került forgalomba. Az Oracle cég ekkorra publikálta saját Oracle*CASE Methodnak nevezett módszertanát, egy kemény információ-központú módszertant, amelynek legerősebb technikája az egyed-kapcsolat modellezése. Mint ilyen, az SSADM 4 verziója is lényegében átvette az Oracle-féle adatmodellezést.

Grafikus editora tehát az egyed-kapcsolat modellezését, a funkciók hierarchiájának készítését, az adatfolyam-ábrázolást és a mátrixmodellezést támogatja.

A szótár felépítése már tartalmazza az akkori fejlettségi szintnek megfelelő elemzési és tervezési információkat: egyedeket (beleértve szub- és szupertípust), attribútumokat és domaineket, egyedi azonosítókat, hivatkozási integritás definiálását, idegen kulcsokat, kapcsolatokat (beleértve a

kizáró vagy kapcsolatokat is), kifejezéseket, funkcióleírásokat, eseményspecifikációkat, adatfolyamrészleteket, adatbázis-objektumokat (tábla, oszlop, tablespace, index, view, fájl definíciók) és modulleírásokat. Mindezen elemzési és tervezési információk tárolása mellett *dictionary management* funkciókat is tartalmazott a verzió, azaz verziókontrollt vagy alkalmazásirendszer-struktúra létrehozását.

A szótár többnyire az akkoriban használatos 5-ös verziójú adatbázisban volt, a Dictionary interfész pedig 2.3-as Forms volt. A szótár-implementálásból már adódik a generált adatbázis és a generált alkalmazás is. Az RDBMS 5-ös verziójára lehetett SQL scripteket előállítani, a generált alkalmazás pedig mindössze a FORMS 2.3 -at jelentette.

Ami a visszafejtést illeti, a létező adatbázissémákból a verzió egy kezdetleges szótár-információt tudott készíteni, a FORMS-alkalmazásokból pedig moduldefiníciót.

A CASE-termék két különálló komponensből — CASE*-Designer (grafikus editorok), CASE*Dictionary (hagyományos szótár interfész, adatbázis és alkalmazásgenerátor) — állt. A CASE*Dictionary mint termék önmagában alkalmas minden információnak a szótárba juttatására, nyilván a grafikus úton előállítottakat egyszerűbb a Designer komponenssel szerkeszteni. Ebben az időben VAX, valamint Unix platformokon működött legelterjedtebben az Oracle CASE-eszköze.

5.0-s verzió

Az 1991-ben megjelent 5.0-s verzió kibocsátását a 6-os adatbáziskezelő és a 3-as verziójú SQL*FORMS megjelenése előzte meg, és a változás mindenképpen a generálás területén volt jelentős. De nézzük végig a verziót vizsgálati szempontjainknak megfelelően!

A CASE*Designer területén nem történt semmiféle változás, talán annyit lehet elmondani a magyarországi piacról, hogy lényegében ebben a verzióban kezdték el használni a grafikus editort, a 4-es verziójú CASE-eladások egy része csak a CASE*Dictionaryre terjedt ki.

A szótár tartalmát illetően a következő fontos bővítések történtek. A megszorítások (Constraint) definiálása, osztott adatbázistervezési információk, stratégiai tervezés objektumai (célok, problémák, sikertényezők, szervezeti egységek stb.) mellett a szótárat tetszőleges felhasználó által definiált objektummal bővítették.

Az Oracle termékcsalád akkori fejlettségi szintjéből adódóan a szótár alatt a 6-os verziójú adatbázis volt, a Dictionary interfész pedig a 3-as Formsot jelentette. Ugyancsak az előzőekből következik, hogy a generálások elsősorban RDBMS 6-ra, Forms 3.0-ra, SQL*Menu 5.0-ra, illetve SQL*ReportWriter 1.1-re történtek. Természetesen az alacsonyabb verziószámú 4GL eszközökre lehetett továbbra is alkalmazást generálni.

A platformot illetően a hangsúly áttevődött a Unixra, az első verziók a Sun gépcsaládra jelentek meg.

5.1-es verzió

Kezdjük a megszokott sorrendben, és a „rossz hírekkel"! A grafikus editorok vonalán változás az ígéretek szerint csak az ez év áprilisában kibocsátott, gyökeres változást hozó termékben következik be. De kellemes, a fejlesztést és a használatot segítő egyéb információkról is beszámolhatunk az 1994-es CASE-verzió kapcsán, amely az Oracle Cooperative Development Environmentjének egy eleme, és mint ilyen, CDE CASE néven is ismeretes.

Az Oracle 7-es adatbáziskezelő új lehetőségeit ez a CASE-eszköz messzemenően támogatja. Élő server constraintek definiálhatók, tárolt procedúrák, csomagok, funkciók és adatbázistriggerek hozhatók létre, fejlett hozzáférési jogosultság definiálható, és új adattípusok alkalmazhatók, osztott rendszer tervezhető (osztott objektumok, adatbázislinkek, snapshotok stb.). Továbbá a teljes grafikus user interfész (GUI) rendelkezésre áll mind az eszköz használata esetén, mind a generált alkalmazásban (Oracle Forms 4.0, Oracle Reports 2.0). Javított Edit text lehetőség is használható (direkt hozzáférés host editorokhoz, PL/SQL parser stb.), és a létező táblák visszafejtése már egyed szintig történhet, valamint sok minőségileg új riport (alkalmazásirendszer-metrika) kérhető. Platformváltás lényegében nem történt, maradt elsődlegesen a Unix-környezet, és ezen belül vezet továbbra is a Sun gépcsalád.

CDE2 CASE

A jelenleg CDE2 CASE-nek nevezett friss CASE-eszköz több, mint egy új verzió. Az egész technológia jelentős kiterjesztését jelenti, s majdnem minden összehasonlítási szempontból jelentősen újat hoz.

Nézzük először a Grafikus editor komponenseit! A régiek az előző verziókon keresztül összegyűjtött hiányosságok megoldását tartalmazzák. De csak a legfontosabbakat említsük! A diagramokat tetszés szerint rendezhetjük el, az egyed-kapcsolat diagramon az attribútumokat, egyedi azonosító komponenseket is fel lehet tüntetni. Az objektumokkal

csoportosan is lehet műveleteket végezni, és a funkcióhierarchia diagramok, mátrixok layoutját tárolni lehet, a mátrixok oszlopaira és soraira szűrőket lehet definiálni stb.

A régi editorok mellett számos új áll a felhasználók rendelkezésére. Ilyen például az új Process Modeller, amely lehetővé teszi, hogy az üzleti folyamatokat ábrázoljuk a stratégiai elemzés során, és a felhasználóval folytatott megbeszélések eredményét minél előbb ilyen módon rögzítsük. Idő- és költségtényezőket állapíthatunk meg, tetszőleges annotációkat fűzhetünk a diagramhoz, valamint multimédia-tulajdonságokkal egészíthetjük azt ki.

A Data Diagrammer a fizikai adatbázistervezés grafikus eszköze. Elsődleges célja az adatbázisséma grafikus ábrázolása. Mint ilyen, táblákat, oszlopokat, nézeteket (view), elsődleges kulcsokat (primary key), idegen kulcsokat (foreign key) és felvételeket (snapshot) kezel, ugyanakkor alkalmas a fizikai megosztás szemléltetésére. Nagyon hiányzott egy ilyen grafikus eszköz a régi verziókból, de használata az egyed-kapcsolat modell elkészítése után ajánlott.

A megerősített fizikai tervezésnek, illetve kivitelezésnek három új eszköze áll rendelkezésre. A Module Structure Diagrammer segítségével a modulhálót grafikusan létre lehet hozni. Természetéből adódóan kezelése sok hasonlóságot mutat a funkcióhierarchia diagrammal. Szintén új lehetőség a Module Logic Navigator, amely PL/SQL modulok grafikus úton történő kódolását segíti elő, a vezérlőszerkezeti elemek grafikus ábrázolásával és kiválasztásával, szintaxisvizsgálattal, valamint az érvényes objektumnevek felkínálásával. A harmadik eszköz, a Module Data Diagrammer lehetőséget ad, hogy a modulok adathasználatát grafikusan definiáljuk vagy szemléltessük. Megjelenítését illetően hasonló a Data Diagrammerhez.

Az új eszközről...

A szótárban való közlekedés és adatkarbantartás eszköze a Repository Object Navigator (RON), amely teljesen kiváltja a régi verziók CASE Dictionary formátumait. Ez egy teljesen objektumorientált grafikus eszköz, amely több alkalmazási rendszer szimultán kezelését is lehetővé teszi. Két fontos módja van: a „navigálási ablak" és a „tulajdonság ablak", amelynek segítségével láthatjuk az adott objektum leíró adatait. Az eszköz használatát a grafikus Preferences Navigator, valamint az online dokumentáció egészíti ki.

A rendszerkészítés nyitó (front-end) fázisában új lehetőségeket nyújt. Ilyen az említett Business Process Modeller komponens. A middle-CASE (rendszerelemzés) vonatkozásában nagyon sokat javítottak a régi grafikus editorokon, és remélhetőleg minden igényt kielégítenek.

A back-end vonalon (a fizikai tervezés, kivitelezés fázisában) egyrészt az időközben bevezetett új adatbázistechnológiai tulajdonságok magas szintű tervezését segíti elő, másrészt a hagyományos implementálási munkák előkészítésében és végrehajtásában nyújt sok új lehetőséget.

Mivel az elsődleges platform áttevődött a PC-re és az MS Windowsra, így az új CASE-eszköz remélhetőleg szélesebb fejlesztőrétegek számára is elérhető lesz.

Bánné Varga Gabriella

INFORMÁCIÓKÉRÉS: 0530 ▲

AKCIÓ!!!

Tel.: 06-20 342-187

PC 486 számítógép 95 000,– Ft + áfától

4 MB RAM / 512 kB VGA / 1,44 FDD / 350 MB HDD

Monitor Memóriaelemek Winchester Floppy DD Processzor VGA megh. CD

Bővítés alkatrészáron

Indulási segítség

INFORMÁCIÓKÉRÉS: 0541 ▲

NS NET+STAR

Számítástechnikai Szolgáltató KFT.

1148 Budapest, Boros Mátyás u. 12/B
Telefon/Fax: 163-5214, 220-7914

- **Számítógépek, számítógép-hálózatok** tervezése, kivitelezése, optimalizálása, javítása, karbantartása
- Irodatechnikai termékek forgalmazása
- Telefon-alrendszerek kiépítése
- HEWLETT PACKARD Canon Microsoft NOVELL Panasonic termékek forgalmazása
- CASIO Szakszerviz

Casio készülékekhez kiegészítők:

AD-1N, AD-5N Casio-kompatibilis tápegységek, PC ↔ CASIO kommunikációs program + interfész

---

Mega, integrált software csomag

UNIX-os háttér, EK-szabvány, client/server és hálózati opciók, online vagy kötegelt feldolgozás, saját vagy idegen adatbáziskezelők pl.: INFORMIX, ORACLE. Rendszer intagrálás szabványos emekből, működő adatbázisok, PC-s hálózatok rendszerbe építése egyedi igények szerint, komplett projekt kidolgozás

Rugalmasan paraméterezhető kereskedelmi rendszer
több pénznem és raktár, tervadatok, lejárati idő, visszamenőleges leltárkezelés, számla archivum, reklamációk kezelése, részszállítások, ügynökök, jutalék számítás, szerződések, hitelképesség figyelés (könyvelési modul alapján), kifizetési terv készítés, automatikus könyvelési köteg átadás, részlegenkénti rendelés feldolgozás, kis- és nagykereskedelem.

Könyvelés, költségszámítás, Pénzügy, Tárgyi eszköz nyilvántartás, Munkaügy, Bérszámfejtés, Automata raktárkezelés, Kommunális szolgáltatások

ACP háttértárolók, Jukebox-ok, CD archiválók Crystal dokumentum archiváló, INFORMIX adatbáziskezelő, UNIPLEX irodai rendszer

QUANTUM Informatika Kft.
1121 Bp. Denevér u. 48/d. T/F: 162-0466
IFABO A Pav. 110/e

INFORMÁCIÓKÉRÉS: 0552 ▲

SolarSoft — mennyiért, hol?

Shareware programok	Árak áfával
5,25" DD lemezen, 1 lemez	399 Ft/db
5,25" DD lemezen, 3 lemeztől	379 Ft/db
5,25" DD lemezen, 5 lemeztől	359 Ft/db
5,25" DD lemezen, 10 lemeztől	339 Ft/db
5,25" HD felár	100 Ft/db
3,5" HD felár	200 Ft/db
5,25" HD katalóguslemez	249 Ft/db

C.Computer Bt
1039 Bp. III., Kabar u. 1. Telefon: (60)334-336

Floppyland Kft
1056 Bp., V., Váci u. 84. Telefon/Fax: 118-2651, 266-8971

INFORMÁCIÓKÉRÉS: 0506 ▲

"CHIP TIPP" csak 87.170 Ft FAX-390DT

Fax-Telefon-Üzenetkezelő



600 dpi LÉZERNYOMTATÓ

HL-660 - "AZ 1994 ÉV NYOMTATÓJA"

- Gyors GDI nyomtató WINDOWS-hoz
- HP LaserJet 4, Epson és IBM Proprinter emulációk
- 6 lap/perc sebesség, 2 MB (max. 10)
- Egyenesvonalú papírvezetés
- Kisméretű papírra is nyomtat
- Magyar kézikönyv, CP 852
- 200-lapos aut. + kézi lapadagoló
- Olcsó üzemeltetés, toner 5.520 Ft

HL-630 300 dpi-s változat is kapható

NYOMTATÓ FAX ÍRÓGÉP CÍMKENYOMTATÓ

brother

MÁRKASZAKÜZLET

DIT DIGITÁLTECHNIKA

Győr, 9024	Budapest, 1149
Mónus I. u. 19.	Róna u. 75.
T/f: 96/414-411,	T: 30/463-657
417-802	T/f: 267-6769/15
	Fax: 267-6768

KEDVEZMÉNYES ÁRAKKAL VÁRJUK AZ IFABO A épület 312/G standján

Brother International Hungary Kft

Színes, öntapadó címkét készíthet

68-féle alapanyag szalag:
- Laminált, vízálló
- Műanyag ill. papír alapú
- Satírozható, vasalható, átlátszó
- 6-9-12-18-24 mm szélesség

Szöveg Jel Grafika Vonal-kód

» Önálló, billentyűzettel, kijelzővel rendelkező készülékek
» Nyomtatás számítógépről

KÉRJEN BEMUTATÓT!

P-touch

INFORMÁCIÓKÉRÉS: 0516 ▲

Kedit for Windows

„Kellemesen csalódtam...”

A Kedit for Windows a jelek szerint felülmúlja elődjét, a DOS-os változatot. Sőt a cikk szerzője szerint ez az elmúlt 2-3 év legkorrektebb, legjobban megírt windowsos alkalmazása. Az ismertetés első fele általánosan mutatja be és teszi „helyére” a programot, majd később részletesen szól a Kedit for DOS-hoz képesti lényeges különbségekről, többletszolgáltatásokról.

Két dolgot szeretnék előrebocsátani. Először is elfogult vagyok a Kedittel szemben, hiszen több mint 8 esztendeje szinte nap mint nap használom, „ő” a kedvenc texteditorom. (Amiben mellesleg nem vagyok egyedül, mert az [Új] Alaplap szerkesztői ugyancsak 6 éve ezzel dolgoznak — mégpedig tőlem teljesen függetlenül ismerkedtek meg vele!) Másrészt nagyon tartottam attól, hogy ez a rugalmas, gyors, kézreálló program megteszi első lépéseit a kimúlás felé, ha elkészül windowsos változata. Vagyis ösztönösen féltettem az „Ablaktól” a Keditet.

Röviden a Kedit múltjáról

A Kedit eredetileg az IBM nagygépes Xedit programozói editorának PC-s megfelelője. A Xedit nem volt még a mai normák szerinti full-screen (azaz teljes képernyős, kurzorral bejárható) editor, így a Keditnek is volt három editálási üzemmódja, ezek mára már gyakorlatilag érdektelenné váltak.

Másik lényeges ismérv volt az ún. parancssori (command line) vezérelhetőség, valamint az ehhez tartozó REXX-szerű parancs- vagy makrónyelv. Többoszlopos, sororientált sorbarendezés. Egy programozó számára gyakorta egyszerűbb és kényelmesebb volt egy komplett, 2-3 paraméterrel egyértelműen meghatározható parancs kiadása, mint menükön, párbeszédablakokon keresztül kommunikálni az editorral. A Kedit for DOS-ban máig sincs menüsor vagy legördülő menü. Az F1-re előugró Help rendszer és az egérkezelés is csak 3 évvel ezelőtt lett része a DOS-os Keditnek.

További alapvető dolog egy texteditornál, hogy blokkműveleteket és oszlopfunkciókat is támogasson, valamint az áttekinthetően tabulált, jól olvasható forrássorok beírásában segítse a programozókat, vagy pl. akiknek a szöveget DTP-rendszer alá kell előkészíteniük.

Az eddig felsorolt tulajdonságok tehát egy programozói vagy texteditor elengedhetetlen követelményei. Lássuk ezek után, hogy a Kedit for Windows mennyiben felel meg ezeknek a speciális szempontoknak, elvárásoknak!

Jobb, mint a két WW vagy a WP

A Kedit for Windows jól ötvözi saját eredendő szolgáltatásait a Windows grafikus kezelői felület adta általános funkciókkal. Ablak, scroll-bar, menüsor (a megszokott File, Edit, Actions, Options, Windows és Help), button-bar alul és felül is (nyomógombok a funkciók azonnali végrehajtásához). A Clipboardon alapuló törlés, másolás, elmozgatás (cut, copy, paste). Drag & drop, azonnali move vagy copy, akárcsak a legkorszerűbb Microsoft programokban. Sor, oszlop és kurzortól kurzorig (stream) tartó blokk-kijelölés, ezen belül jobbra, balra vagy középre igazítás, kis- és nagybetű konverziója (a speciális ékezetes karakterekre is!), eltolás mindkét irányba, kitöltés adott karakterrel stb...

Nagyjából ennyi az egész. Egyszerű, és mégis nagyszerű.

A Windows Write-tal, a WinWorddel vagy pláne a WordPerfect for Windows-zal még akkor sem lehetne ezeknek a funkcióknak egy részét végrehajtani, ha fix szélességű betűket tartalmazó fontkészlettel dolgoznánk (pl. Courier). Egyébként a KeditW installálás közben kikeresi a Windowsunk alatt pillanat-

nyilag meglévő fix méretű (ún. fixed pitch) betűkészleteket. Ezek is lehetnek TrueType fontok, de az oszlopfunkciók koordinátahelyes végrehajthatóságának alapfeltétele, hogy minden karakter azonos méretű legyen.

Kiknek a legjobb?

Tulajdonképpen ez az egyetlen igazán lényeges eltérés a „menő" szövegszerkesztőkhöz viszonyítva: mindig csak egyetlen kiválasztott fonttal dolgozhatunk. Ezért azután a Kedit for Windows-zal csak úgy levelezhetünk, mint egy hagyományos írógéppel. De ez nem is baj.

Mielőtt áttérnék néhány finomságra, amelyek az ínyenceket és a Kedit for DOS ismerőit érdeklik majd elsősorban, itt jegyezzük meg, hogy a KeditW viszonylag kicsi (mérete kb. 560 kbájt), igen gyors és ügyes program, teljes körű Helppel (1,4 Mbájtos a helpfájl), amihez már csak egy KEDITW.INI fájl kell a Windows könyvtárában, és tulajdonképpen ennyi az egész.

Minden olyan Windows-felhasználónak melegen ajánlható, aki gyorsan akar gépelni, gyakran cserél szöveges adatot DOS-os és windowsos környezet között (automatikus konverzió kérhető!), nagy mennyiségű adatot rögzít DTP-hez, vagy pedig programoz.

Csábít az átállásra

A DOS-osoknak nem mindegy, hogy átállnak-e vagy sem. Figyelmükbe a következőket ajánlom:

— Használható az összes régi billentyűkiosztás (bár javaslom — még a legmegrögzöttebb Windows-ellenes kollégáknak is —, hogy az első két napban mindenképpen az új, windowsos, azaz CUA-felületet próbálják ki, mert érdemes), a régi .KML-ek egy az egyben beolvastathatók.

— Az IBM ASCII 256 karaktere (Windows-terminológia szerint ez az OEM-készlet) és a Windows ANSI karakterek egyaránt használhatók. A speciális karakterek bevitele ötletes: Alt + szám az OEM, Alt + 0 + számbillentyűk a kért ANSI kódkiosztás szerinti megjelenítést eredményezik a megadott karakternél.

Az persze lehet, hogy egy adott kódhoz az éppen kiválasztott fontkészletben nincs semmilyen kép megadva, de a Kedit for Windows természetesen ekkor is beteszi a megadott kódot a fájlba. Betöltéskor és kimentéskor kérhető ún. OEMtoANSI vagy ANSItoOEM konverzió.

— A KeditW-ben a nem állandó (non persistent) blokk-kijelölés a standard, azaz ha a kijelölt blokkon kívülre megyünk (egérrel vagy kurzorral), akkor a blokk-kijelölés megszűnik. Ugyanígy a kijelölt blokknál a gépelés a blokkot azonnal kitörli, és a helyére kerül.

— A nyomtatás a Windows Print Manageren keresztül zajlik, de csak a teljes fájlszintű nyomtatás indítja el azonnal a nyomtatót. Lehetőség van soronkénti nyomtatásra is, erre szolgál a PRINTER CLOSE parancs (így kerülhetnek egy lapra a külön kiválasztott, nyomtatásra időben nem egyszerre kiadott sorok).

— Syntax coloring. C/C++, REXX és KEXX forrásprogramok automatikus tabulálása és kiemelőszínekkel történő megjelenítése a beépített parserrel, amely a Kedit parancsok beírásakor azonnal formai ellenőrzést végez.

— Reguláris kifejezések használata. (Keresés, csere vagy target megadása.) Sokrétűsége alapján ez az opció akár külön cikket is megérdemelne. Röviden összefoglalva: logikai összefüggéseket is figyelembe vevő dzsókerekkel, nagyobb halmazokkal, szűkítésekkel, kizárásokkal, indirekt hivatkozásokkal megspékelt, nagyon rugalmas hivatkozási mód.

— A beépített nyelv immár a lebegőpontos aritmetikát is támogatja!

— Alacsony szintű, fájl-, karakter-, illetve sororientált funkciók: LINEIN(), LINEOUT( ), CHARIN( ), CHAROUT( ), LINES( ) és CHARS( ).

— Az ITERATE, LEAVE ... END ciklusszervező parancsokhoz változót is rendelhetünk.

— A KEXX makró legnagyobb mérete 4000 sor, egy sor maximálisan 256 karakter hosszú.

— Újabb, definiálható billentyűkombinációs lehetőségek, például Shift-Ctrl-... vagy Alt-Ctrl-... stb.

— A SORT parancs fizikai méretbeli határai is megszűntek. (Ami a DOS-os Kedit-hívők régi vágya volt.)

— Kis- és nagybetűre érzékeny csere (case-sensitive replacements).

— Windows-specifikus parancsok: DIALOG, WINEXEC, CLIPBOARD, MERGE. POPUP, SHOWDLG, WINDOW, WINHELP, SET DRAG, SET INTERNATIONAL, SET TOOLBAR, SET TOOLBUTTON stb.

— A KeditW belső fizikai határai:
Egyszerre megnyitható fájlok maximális száma: 40.
Maximális sorhosszúság: 4096.
Fájlméret: korlátlan.

— WinWord-szerű UNDO és REDO.

— Command, Find/Replace, QuickFind history.

— A már jó ismert szelektív editálás (ALL...) mellé behozták az eltérő színnel történő kijelölést is (TAG...).

A program ára: 26 000 Ft + áfa, de szerintem ér is annyit! És ahogy azt az olvasó időközben valószínűleg kitalálta, ez a cikk már Kedit for Windows-zal készült, sőt feltehetően a többit is azzal fogom írni.

Ha tehát írásaim ezentúl jobban sikerülnek, mint az eddigiek, rögtön tudhatják az okát...

Herczeg József

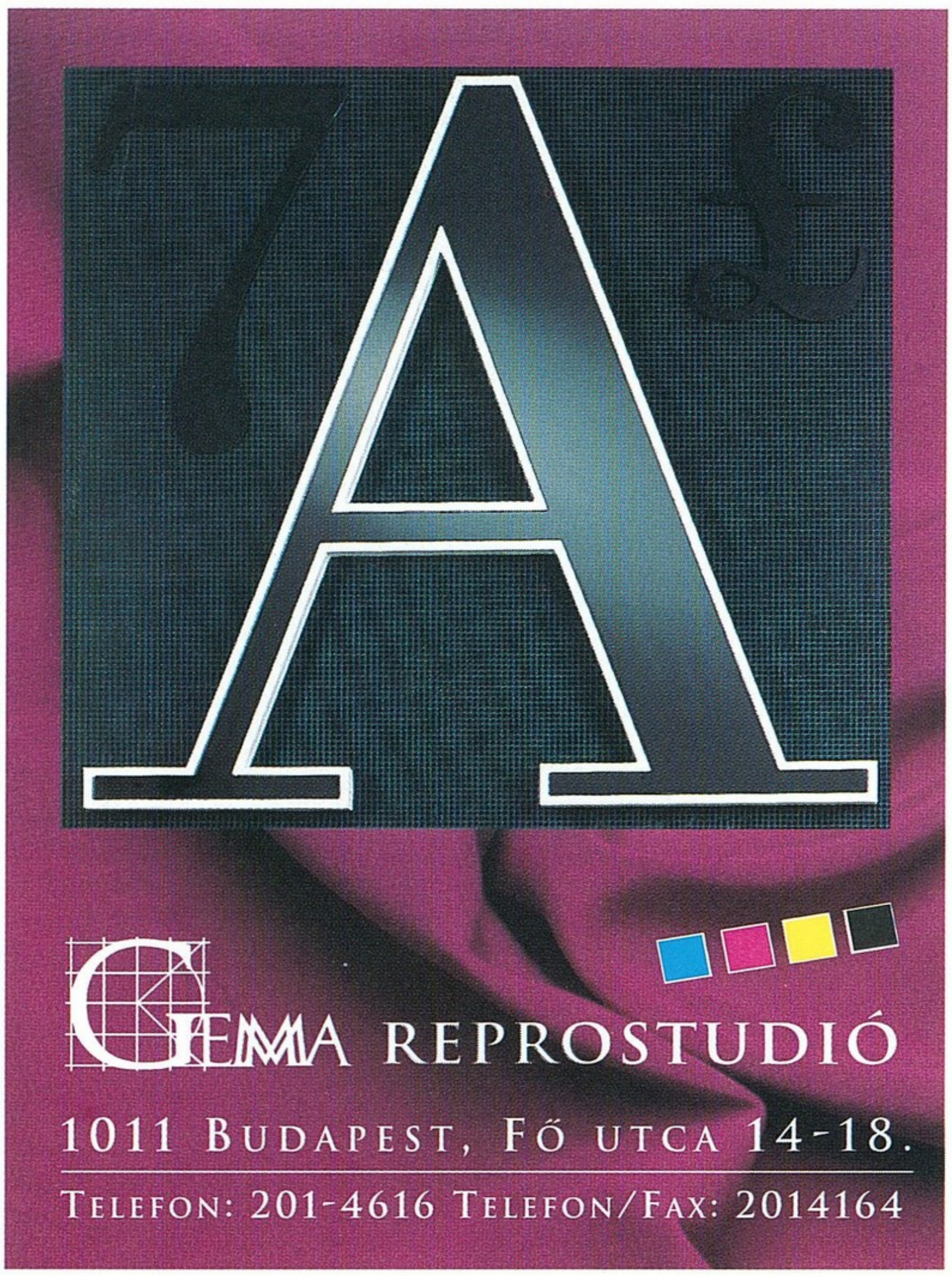
SPIELER KFT.
COMPUTER

1083 BUDAPEST, ILLÉS U. 40.
Telefon/Telefax: 134-3715 • Telefon: (60) 325-351
Nyitva: 9.00–12.00 és 14.00–18.00 óráig

DESKTOP deLUX HÁZ Műanyag ház teljes belső árnyékolással, alacsony zajú tápegységgel	13 600,– Ft
BABY deLUX HÁZ green	7 200,– Ft
MINI TOWER HÁZ **kijelzővel, 18 hónap garancia a tápegységre**	5 700,– Ft
486DX-256K **cache, 4x72 pin ram foglalat, AMI Bios, GREEN ALAPLAP**	14 300,– Ft
VGA COMBO kártya **CL.5434 1MB (max 2MB)+ gyors IDE PLUS 2SPG**	13 890,– Ft
14" ***ACER*** P.I. SVGA LR NI color green monitor	34 300,– Ft
14" **TATUNG** SVGA LR NI color green monitor	34 000,– Ft
15" ***ACER*** P.I. XVGA LR NI color green monitor	49 200,– Ft
17" ***ACER*** P.I. XVGA LR NI color green monitor	94 400,– Ft
RICOH 4630 A4-A3 fénymásoló	172 300,– Ft
POWER SAVER kártya **286–486-ig minden PC-ben használható GREEN eszköz szilárdtest relével, kikapcsolja a MONITORT, PRINTERT stb., amíg ÖN mással van elfoglalva, az EPA mérései szerint 65% energiát megtakaríthat**	5 900,– Ft
COMPUTER REMOTE SYSTEM **3 db PC-ről programozható vagy kézi távkapcsolóval működtethető 220V dugalj + SW**	11 400,– Ft
TELETEXT kártya **teletext adások a PC-n, információk bármelyik csatornáról, lapozható, rögzíthető stb., DOS és WIN alatti kezelő SW-rel**	15 900,– Ft
QTRONIX SCORPIUS magyar szabvány szerinti BILLENTYŰZET	2 280,– Ft
QTRONIX LYNX-30 3 gombos MOUSE 6400 dpi, SW + mouse pad + mouse-tartó	2 050,– Ft
UPS, ami házba szerelhető 300- vagy 500W	**28 500 v. 30 800,–** Ft
MASTERDATA 5.25" DS/HD floppylemez 10 db/doboz	470,– Ft
100 db/doboz	4 500,– Ft

Áraink az ÁFÁ-t nem tartalmazzák. Az árváltozás jogát fenntartjuk.
Vám- és áfamentes beszerzés bonyolítását is vállaljuk!
Kitűnő parkolási lehetőség.

INFORMÁCIÓKÉRÉS: 0560 ▲



INFORMÁCIÓKÉRÉS: 0523 ▲

CORWELL Kft.
1143 Budapest, Utász u. 5.
Tel./Fax: 251-9831, 252-4359

Tisztelt Partnerünk!

Nekünk Ön a legfontosabb! Ezért cégünk fennállása óta arra törekszik, hogy a számítás- és irodatechnikai kellék termékkörben nálunk mindent megtaláljon, amire szüksége lehet.

Ezért bővítjük folyamatosan termékválasztékunkat. Ezért lettünk magyarországi disztribútora a világ vezető gyártóinak és tartjuk állandóan raktáron termékeiket. Néha egy-egy termékkörben több nagy márkát is.

Most, az IFABO idején szeretnénk számot adni Önnek arról, hogy céljaink megvalósításában hol tartunk. Kérjük, látogassa meg kiállításunkat május 9–13. között az „A" pavilon 311/i standján, hogy bemutathassuk Önnek a kínálatunkban szereplő termékeket.

Verbatim – mágneses adathordozók

office data – PC-tartozékok és bútorok

PRIMAX – egerek, szkennerek, kábelek, multimédia és iratmegsemmisítő

3M – mágneses adathordozók

GRAHAM MAGNETICS – mágnesszalagok és streamerkazetták

ESTECE – irodaszerek

Victoria – leporellók és faxpapírok

APLI – számítógépes etikettek és fóliák

Reynolds – írószerek

moLin – írószerek

EMBATEX – írógépszalagok és printerkazetták

Fujitech – írógépszalagok és printerkazetták

Prince Moran – irodai székek

ESSELTE – oktatási szemléltető eszközök

Post-it™ – irodaszerek

ORIGINAL – és for use tonerek és tintapatronok

CORWELL

Nekünk Ön a legfontosabb.

U.i. Országos hálózatunk bővítéséhez viszonteladók jelentkezését várjuk.

INFORMÁCIÓKÉRÉS: 0512 ▲

PADS-PowerPCB

A gördülékeny tervező

A PADS rendszerek már első változataik megjelenésétől ott vannak az élvonalban. A nyomtatottáramkör-tervező szoftver neve napjainkra már jól cseng hazánkban is. A sikerhez tartozó érdekességekről, újdonságokról folyamatosan beszámoltunk, és a mostani „jelentés" apropója szintén hasonló: tavasszal ismét vadonatúj dolgokkal rukkol(t) ki a rendszerek fejlesztője és gyártója, a PADS Software Inc.

Akik kezdettől fogva — Magyarországon 1991 óta — figyelemmel kísérték a PADS rendszer fejlődését, tudják értékelni a bizonyítékot: 1994 végén jóval több mint 20 000 felhasználó szerepelt a PADS Software Inc. regisztrációs listáján.

A PADS cég elsősorban annak köszönheti népes felhasználói táborát, hogy a legújabb programozási tecnikák alkalmazásával rendkívül rövid idő alatt kifejlesztette és integrálta azokat a modulokat és programrészeket, amelyeket a felhasználók vártak. A felmerülő igények egyrészt a technológiafejlődés következményei, másrészt megfogalmazzák azt a jogos elvárást, hogy a szoftver maximálisan használja ki a mindenkori legmodernebb hardvereszközök lehetőségeit.

Néhány dolog, amiben a PADS élen járt:

— A PADS volt az első olyan sémaeditor, amelyben a több lapból álló kapcsolási rajzok valamennyi lapjának minden információja folyamatosan a tervező rendelkezésére állt, nem csak az éppen használt lap részinformációi.

— A 386-os gépekkel szinte egy időben, elsőként jelent meg a PADS rendszerek 32 bites változata (szoftver és adatbázis), az első integrált Push&Shove algoritmusú autorouter stb.

— A PADS volt az első igazán hardverfüggetlen nyomtatottáramkör-tervező rendszer. PC-kre (DOS, Windows, NT) és az összes Unix-munkaállomásra (Silicon Graphics, Sun, HP stb.) írt változata ugyanúgy néz ki, és az adatbázisok konverzió nélkül hordozhatók a különböző platformok között.

— Rendkívül tág határok között konfigurálható rendszer, így a felhasználó pontosan azokat a szolgáltatásokat vásárolja meg, amelyekre valóban szüksége van.

Új irányt szab

Az elektronikai tervezőrendszerek „diktátora" 1995 márciusában ismét színre lépett, és forgalomba hozta a PADS-PowerPCB-t, amely valószínűleg „felpörgeti" az egész EDA (Electro Design Application) iparágat. Tekintsük át röviden a változások közül már az első látásra is igen fontosnak tűnőket.

Átírták az alapszoftvert (objektumorientált C-be), és ezzel párhuzamosan létrehozták az objektumorientált adatbázist. Tették mindezt a még gyorsabb futásidők érdekében. Ezzel olyan új kényelmi modulok (például a CLI: Command Language Interface) beépítése vált lehetővé, amelyekkel a felhasználók egyszerűen és gyorsan férhetnek hozzá az objektumorientált adatbázishoz.

Vadonatúj modul az IPC (Inter Process Communication), amely lehetővé

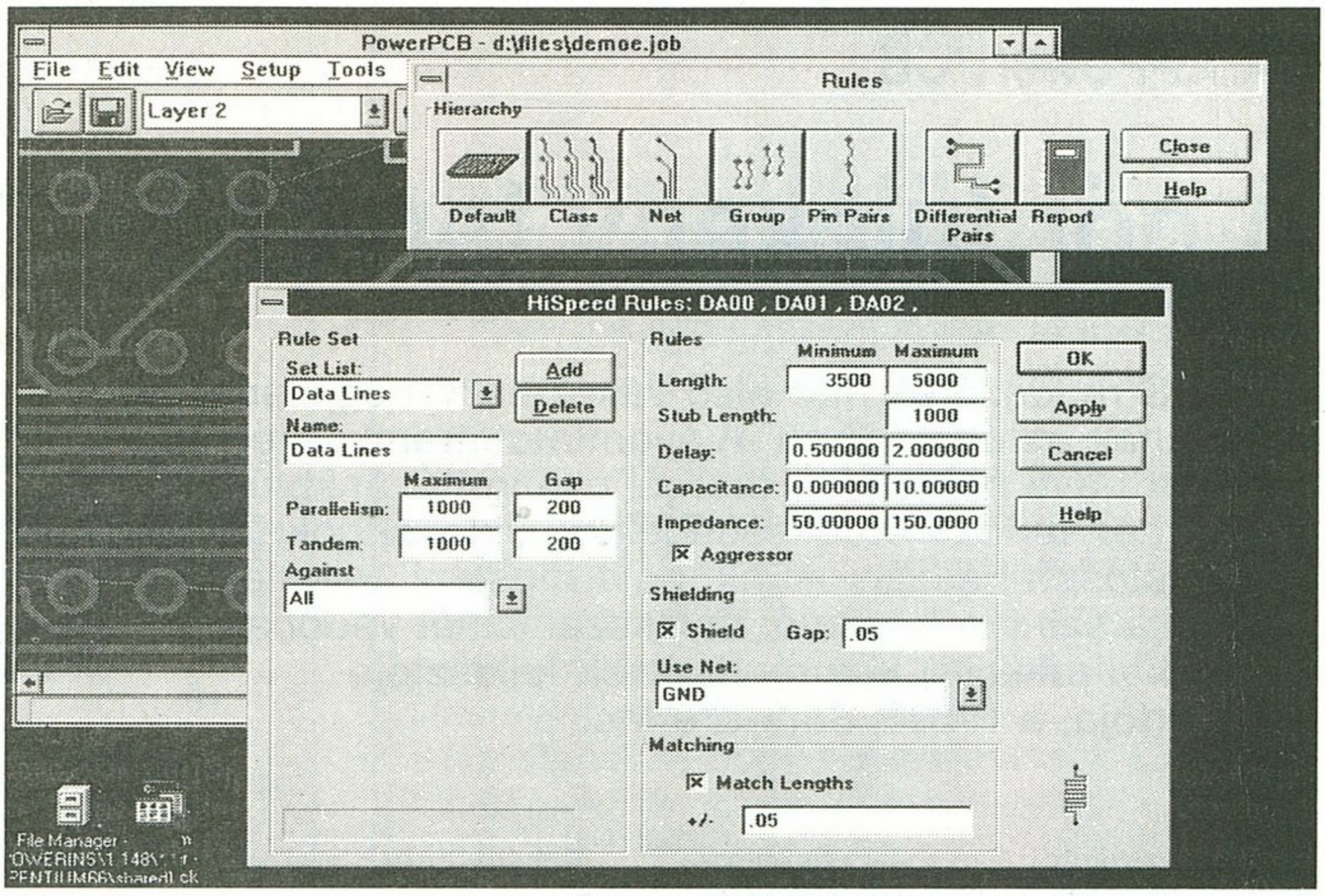
teszi a keresési, elhelyező, huzalozási és ellenőrzési funkciók csatolását a sémaeditorhoz (PADS-Logic — saját sémaeditor; PADS-View — a Viewlogic-tól megvásárolt kód). Az IPC használatának szép példája, hogy az elhelyezéskor együtt tartandó elemek csoportokba sorolása már a kapcsolási rajzon megtörténhet. Így az intelligens elhelyező program (Cluster Placement: CPR) behívásakor már helyből rendelkezésre állnak azok az alapvető információk, amelyek a CPR működéséhez szükségesek.

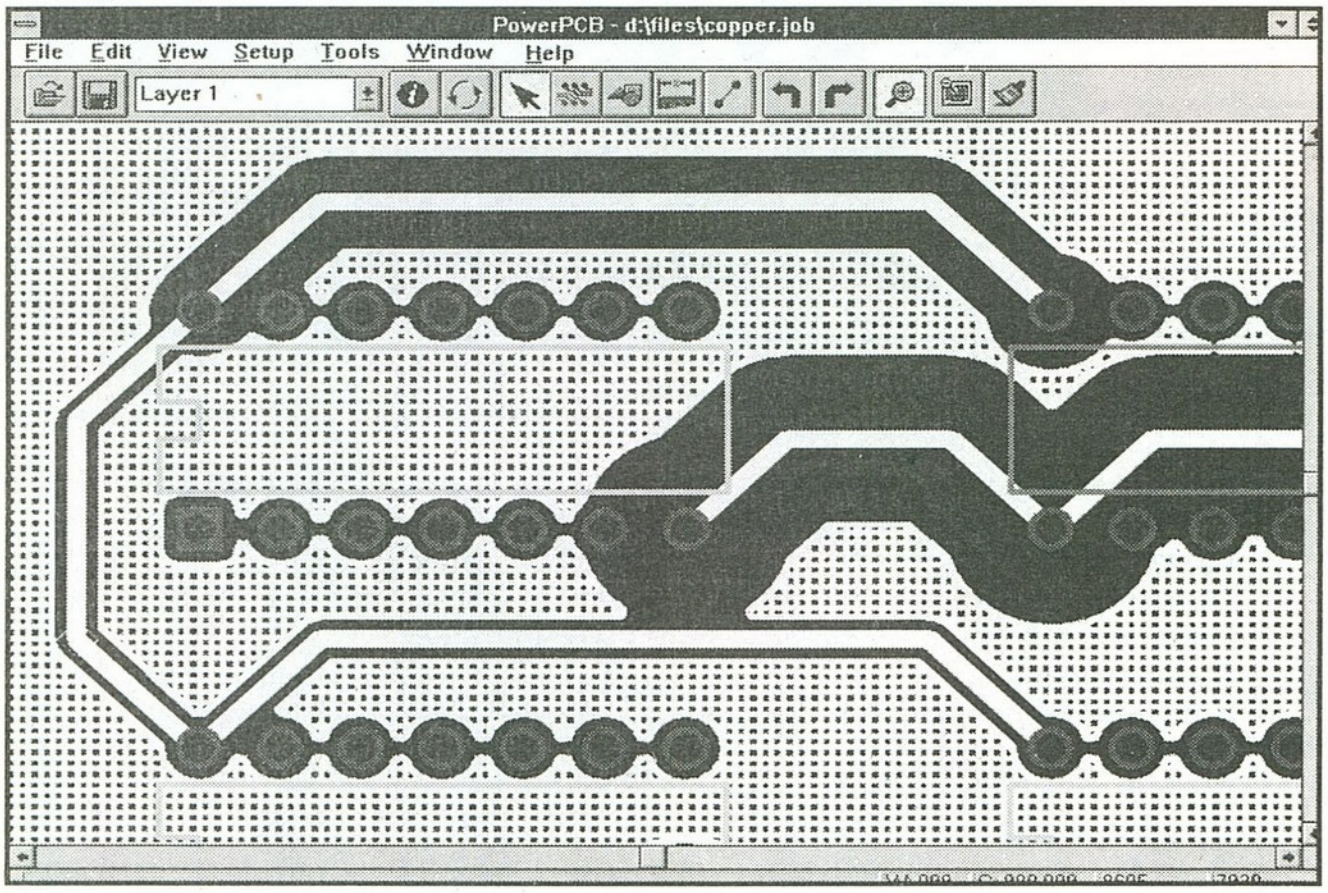
Lehetséges a „forma alapon” tervezés (Shape Based PCB Editor). Ez teszi lehetővé a teljes elszakadást a gridtől az elhelyezés és a kézi vagy automatikus huzalozás során. Mindez biztosítja a tervezés alatt álló kártya felületének lehető legtökéletesebb kihasználását, illetve az adatbázis felbontásának lecsökkentését (tetszőleges mértékig).

A DRE (Dynamic Route Editor) a kézi huzalozást segíti az akadályok automatikus eltolásával a huzalozás útjából. A vezeték megrajzolásakor minden olyan objektumot arrébb lök, amelyről úgy döntünk, hogy útban van, és eltolása lehetséges.

Említést érdemel még a CCT-link, amely a Cooper & Chyan Technology Inc. mind népszerűbb Spectra routerének integrálását teszi lehetővé. A PADS és a CCT között eddig is szoros volt a kapcsolat, de ilyen mértékű integráltság még nem valósult meg. Érdekes még a DRC (Design Rule Check) többmódú alkalmazása. Online alkalmazás melett csak olyan huzalozást hozhatunk létre, amely megfelel az előre meghatározott szabályoknak. Utólag batchben futtatva kétféleképp használhatjuk: vagy a hibahelyeket listázzuk és jelenítjük meg, vagy — a hibákat felismerve — a DRC rutin megszünteti azokat. Érdekes még az automatikus méretező modul, amely a kártya fizikai méreteit adja meg a technológia számára, beleértve a lecsapások, lekerekítések, ívek paramétereit is.

Kifejezetten a felhasználók kívánságára került a rendszerbe az online help, és a több mélységű Undo-Redo funkció. Továbbfejlesztették az EDC (Electro Dynamic Checking) modult, a fóliafolyatással kapcsolatos funkciókat, és még számos más részt.





„Értelmes” kezelői felület

A PADS menürendszere eddig a hierarchikus felépítést követte. Bármennyire redundanciamentes is volt ez a menü, a sok modul és az emiatt folyamatosan szaporodó funkciók sokasága miatt a nagyobb PADS-konfigurációkban esetenként 4-5 hierarchikus szinten keresztül lehetett elérni egy-egy keresett menüpontot, majd ugyanennyi szinten át visszatérni oda, ahonnan elindultunk. Ezért a PADS Software Inc. úgy döntött, hogy megszünteti a hagyományos értelemben vett menüt. A menü ezentúl már nem hierarchikus szervezésű, sőt alapállapotban nem is látható a képernyőn.

A rendszer felméri a szálkereszt helyét a kártyaterven, és ezen a helyen legördít egy vagy több ablakot, amelyben azok és csak azok a funkciók láthatók, amelyeknek az adott esetben egyáltalán értelmük lehet.

Lássunk erre egy példát! Tegyük fel, hogy a szálkereszt egy alkatrészláb közvetlen közelében van. Az egér gombjának lenyomása után legördül egy két részre osztott menüablak. Ez azokat a funkciókat (mozgatásokat) tartalmazza, amelyeket az alkatrészen el lehet végezni, illetve amelyekkel az adott lábról huzalozni lehet. Így nagyban leegyszerűsödött a program használata: csak az van a képernyőn, aminek „értelme van”, és ez a betanulási időt is jócskán csökkenti.

A PowerPCB magyarországi bemutatkozása az Ifabón lesz, és az érdeklődőknek a disztribútor CADserver tesztpéldányokat is rendelkezésükre bocsát.

Visi Dezső

Irix, a „szilikonos" Unix

A képzelt valóság motorja

Az egyik felkapott új szakkifejezés a virtual reality. Kevesen tudják, hogy az új terminus technicust a kaliforniai Stanford Egyetem kutatóiból alakult Silicon Graphics Inc. (SGI) cég indította útjára, amely a magyar piacon is — elsősorban a tudományos és a műszaki tervező szférában — gyökereket eresztett. Gyári Unixokat bemutató sorozatunkban most az SGI kerül terítékre.

Az Irixet kezdettől fogva az AT&T Unixra alapították, úgy, hogy beledrótozták a fejlesztésekhez legszükségesebb BSD-részeket is. A MIPS-sorozatra az SVR3-as került, a legújabb gépmodelleken azonban már SVR4-es kód fut. Ez a tény az SGI cég gazdagságára utal. Ugyanis más gyártók húzódoznak az SVR4-es kód vásárlásától, mert eladásonként igen jelentős licencdíjat kell fizetniük a Unix-gazdának, most épp a Novell USG-nek. Jelenleg az SGI hardvereken nem mindegyik Irix-változat használható, már csak a két utolsó változat van forgalomban: az 5.3-as a MIPS 4000-es sorozatú gépekre, a 6.x-es a legújabb modellekre. Az 5.2-esből továbbvitt 6.x-es változat már 64 bites, beleértve az állományokon belüli rekordcímeket is, ami hatalmas állományok kezelését teszi lehetővé. (A Digital Alpha AXP OSF/1 kapcsán olvashattunk ilyesmiről, tehát nem egyedi eredményről van szó, de nem is nagyon elterjedtről.)

Az SGI arra törekszik, hogy tető alá hozza az egységes Unix-platformot. Nem tervezi más konkurens gyártók Unixának (például Sun Solaris) beengedését gépeire. A MIPS-es Unix-változattal való kísérletezést meghagyják a Siemensnek. Közönyösen figyelik a Windows NT körüli erőlködést is, maguk nem áldoznak rá különösebben.

OpenGL

Mielőtt az Irix egyéb paramétereinek a részleteibe bonyolódnánk, ki kell emelni az OpenGL, igen nagy felbontású, 3D grafikás nyelvet, amely már szinte ipari szabvánnyá nőtt. Más Unixok is támogatják, hogy kompatibilisak legyenek az SGI szoftvereivel és adatformátumaival. Az OpenGL egyébként a korábbi Iris GL-ből alakult ki, amelynek átvezetése az OpenGL vonalra éppen most van folyamatban az Irix-változatokban és az SGI-szoftverekben, noha nem nagy a különbség.

Az OpenGL nem ablak- és operációsrendszer-függő, sőt független a hálózatoktól is. Kliens/szerver üzemmódban sem szükséges, hogy kliens és

A teljesítmény megszállottjai

1980-ban azért vált ki Amerikában egy háromfős csapat az egyetemről, mert a katonaságtól kapott megbízást: dolgozzanak ki repülőgép- és más harceszköz-szimulátorokhoz 3D-s grafikai rendszereket. Ez a munka nyilván jól jövedelmezett, mert a cég rohamosan fejlődött. Eleinte Motorola processzorokat használtak grafikai gyorsítóval ellátott gépeikben, de később teljesen áttértek a MIPS cég RISC processzoraira. Közben a MIPS elkövetett egy-két hibát (talán nem kellett volna a CPU-gyártás mellett a gépgyártásba is beszállni), s emiatt a csőd szélére került.

A Silicon Graphics nem engedhette meg, hogy CPU-bázisa megrendüljön, ezért — már akkor is elég tőkeerős lévén — felvásárolta a MIPS-et, szabad kezet hagyva a cég eszének, John Mashey-nek. (Az említett úr minden fórumon köszörülte a nyelvét a többi processzorgyártó bajain. Azóta Mashey helyén más áll, de stílusa öröklődött.)

A Silicon Graphicsnak erősségévé vált az integrált CPU-gyártó részleg. Az R8000-esnek ugyan még nem minden paramétere jelent világcsúcsot, de a csúcstartó DEC Alpha chipek egyes típusaiból gyártott példányszámok kicsit elméletivé teszik azok technikai fölényét. A most kifutó R10000-esek 600 megaflopsos csúcsa igencsak kiemelkedő. Ugyanakkor az R10000-eseknek elsősorban nem a lebegőpontos, hanem az egészaritmetikás teljesítményük nő. Ráadásul még csak 75 MHz-es processzorról van szó. Mi lesz, ha az órajelét sikerül „felcsavarni" 300 MHz-re? A processzorok kapcsán fontos megjegyezni, hogy az SGI-nél már régóta megoldott az SMP (szimmetrikus multiprocesszoros) kezelés.

Az 1985-ben tőzsdére került SGI termelési volumene ma már nagyobb, mint a Suné, az Apple-é vagy a gyengélkedő Digitalé. A 750 millió dollár tőkéjű cég az utóbbi években 42%-os növekedést produkált, de 1995-re már 50%-os növekedést prognosztizálnak, s mindezt 5 000 dolgozóval. A gyorsabb növekedés szükségessé teszi a létszámbővítést: mintegy 2 500 embert vesznek fel, miközben más cégek elbocsátásokra kényszerülnek. A jelenlegi 2,4 milliárd dolláros évi termelési volument 2000-ig 10 milliárd dollárra szeretnék emelni.

Az SGI alapítóinál a 3D grafikás gépek operációs rendszereként csak a Unix jöhetett számításba. A katonaság mellett e gépek iránt leginkább érdeklődő egyetemek is ezt kívánták. Egyébként sem volt tanácsos kifejleszteni egy bonyolult grafikus szoftvert olyan operációs rendszerre, amely nem nyitott a világra. A hardveres hordozhatóságot mindeddig csak a Unix produkálta széles ipari méretekben.

szerver oldalon azonos gép vagy operációs rendszer legyen. (Vigyázat, a .gl vagy .GL kiterjesztésű állományok nem OpenGL állományok!) Az OpenGL licencét egy sor jelentős cég (AT&T, Cirrus Logic, Cray Research, Digital, Evans&Sutherland, Harris, Hitachi, IBM, Intel, Intergraph, Kendall, Kubota, Microsoft, Miro, NEC, Portable Graphics — Sun-OS, HP-UX —, Samsung, Sony, Univel) vásárolta meg, ezzel az OpenGL a piac mintegy 95%-át uralja.

Bombabiztos

Tudjuk, a katonaságnak nem lehet akármit eladni. Egy kutatói környezetben működő Unix „feldobhatja a talpát", ritkán származik belőle nagyobb baj. Azonban egy rakétavezérlő rendszerben ilyen lezserség aligha engedhető meg. Éppen ezért az SGI inkább maga tart fenn erős Unix-fejlesztő gárdát, nem kapkod a piacon kapható, kevésbé szilárd kommerciális unixok után.

Az Irix bizonyos ritka példányai a DOD parancskönyv egészen kemény biztonsági előírásainak is megfelelnek. A B1-es szigor azonban hamarosan bekerül a szélesebb körben terjesztett Irix-változatokba is. Az Irix egyébként többé-kevésbé a ma elterjedt összes Unix-szabványnak megfelel, azonban még elég távolinak tűnik az X/Open Spec 1170-es változatban definiált egységes Unixtól. Az SGI-nek nem is célja, hogy operációs rendszere minden létező Unixszal kompatibilis legyen. Ennek az az oka, hogy gépeiket eddig olyan speciális területeken használták, ahol nem volt széles körű szoftverválaszték. Míg a korábban szintén csak a tudományos és a hadi szférának gyártó Sun a tömegtermelés felé fordult, addig a Silicon Graphics máig megmaradt a tudományos szférában. (Az első eltérést ettől a koncepciótól az jelentette, amikor a tömegigények kielégítésére szánt Indy multimédiás gépsorozatot piacra dobták. Valószínű azonban, hogy ezt a koncepciót alapjaiban rengeti meg a Nintendo céggel nemrég kötött stratégiai megállapodás.)

Reality Engine-2

A képzelt valóságból egy szűkebb közönség már 2 éve megcsodálhatott a Postás Kultúrotthonban egy kis show-t, amelyet az egyik SGI-disztibútor, a Creative Engineering tartott. (Ugyancsak két éve a másik disztribútor, a CADserver rendezett hasonlót a városban.) A postásoknál tartott rendezvényen mindenki láthatta, hogyan készült Hollywoodban a Csillagok háborúja, és a többi tudományos fantasztikus film, a hús-vér szereplők és a computerbábuk keveréke. Azonban már megszületett a totálisan elektronikus film is, a TRON.

A Reality Engine-2 bemutatót leginkább úgy képzelhetjük el, mintha egy léggömbön ülő légy perspektívájából néznénk a világot. A lágy szellő ide-oda visz bennünket egy atomkatasztrófa-riadó miatt teljesen kiürült városkában. Úszunk a házak között, nézzük a forgalom híján céltalanul működő utcai lámpákat, Le Sage módjára leemeljük a házak sipkáját, hogy beléjük nézzünk. A Reality Engine grafikus gyorsítójának hatását jól mutatta, amikor váltogatva néztük a drótváz képet és a masszív alakzatokat. Semmi akadozás vagy villódzás, az átváltások simák. A képgenerálást ugyanis nem a fő processzor végezte, hanem a Reality Engine-2. Azután ködös zónába kerültünk, a színek automatikusan kifakultak a ködfoltok mentén. 70 Mbájtnyi mintaállományban bóklásztunk, amelynek végigjárása akár órákig is eltartott volna. Az időjárási paraméterek változtatásával a látvány variációinak száma elképzelhetetlenül nagy lett.

A bemutató óta a Műegyetemre is bekerült egy képzeltvalóság-gép, a hallgatók már ezen tanulhatnak. Más felsőoktatási intézményekben is kedveltek a Silicon Graphics gépei. Megindult ipari alkalmazásuk is (Olajterv), és fontos szerephez jutottak például a reklámfilmek készítésénél (MTV).

Zsadányi Pál

A játékpiac nem játék

A Nintendo a 80-as évek végén tömegesen tette tönkre az amerikai játékszoftver-kiadókat. (Az egyikben mi, magyarok is érintettek voltunk a Novotrade Játékstúdió révén.) Ugyanis az azóta tönkrement Commodore cég óriási népszerűségre szert tett Commodore 64-eseit otthoni félkarú rablókká butította le azzal, hogy egy rendkívül kompakt megoldással a kezelést a végletekig leegyszerűsítette. Az alapgépbe mindössze egy kazettát kellett bedugni, az egészet csatlakoztatni egy színes tévéhez, és kész volt a konfiguráció. Később más extrák is jöttek hozzá.

A Commodore képi megjelenítését az eredetihez képest az ún. BLIT műveletes hardvermegoldással gyorsították fel. Emiatt a módszert kidolgozó cég beperelte az amerikai Nintendót, méghozzá sikerrel. A történet érdekessége, hogy a felfedező kis cég akkorra már éppen csődbe ment, mire a Nintendo használni kezdte megoldását. A bírósági ítélet indoklásában az is szerepelt, hogy egy már padlón lévő partner találmányát sem lehet büntetlenül eltulajdonítani. A Nintendo csak kicsit prüszkölt a kártérítés összege miatt, hiszen a védettség épp a per eredményhirdetésekor járt le, így tovább már nem kellett jogdíjat fizetnie.

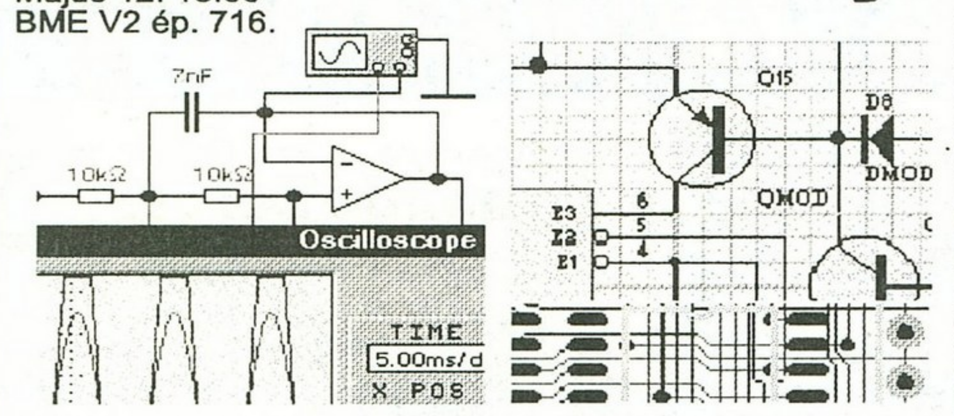
Az amerikai játékszoftverpiacot egyébként a Nintendo azzal tarolta le, hogy monopolizálta a régi játékok átírását Nintendóra. Ehhez az óriási munkához rengeteg olcsó japán programozót állítottak csatasorba. A piac pedig óriásira nőtt: tavaly mintegy 30 000 000 készüléket adtak el (az éves PC-eladások tehetnek ki ekkora mennyiséget), az eladott játékkazetták száma pedig ennek sokszorosa.

Ebben a helyzetben a Silicon Graphics stratégiai szövetsége a Nintendóval bombahír. A Silicon Graphics leányaként működő MIPS CPU-chipgyártó cég máris évi 10 000 000-s szériaszámú CPU-eladásra számíthat: a katonai ipar után már az autóiparba (BMW) és a közszükségleticikk-iparba (mikrohullámú sütők vezérlője stb.) is betörtek. A nagy sorozatok pedig jótékony hatást gyakorolnak az árakra. Ha az Apple—IBM—Motorola triumvirátus annyira örül az egymilliomodik PowerPC-alapú gépének, akkor a MIPS/SGI mit szóljon!?

A Nintendónak az SGI adja a totális 64 bites, 3D grafikás szoftvert. A Nintendo-kapcsolat a szórakoztatóipari alkalmazások új, óriási üzletágába vezeti be az SGI-t. Ide tartozik a „video on demand", és az „interactive video", amelyhez hálózat is kell. Az Indigók, Power Challenge-ek és Onyx szerverek már eleve hálózatos kiépítéssel készülnek, az X-grafika enélkül aligha lenne használható.

Bemutató:
Május 12. 15.00
BME V2 ép. 716.

Electronic Design



Kérje szoftver katalógusunkat!

Sagax Kft. 1093 Budapest Pipa u. 4.
Tel:215-0082, FAX:216-4019

Egyedülálló szolgáltatásokat nyújt a

LEKTOR WINDOWS 4.1

Toldalékoló kivételszótárral rendelkezik.

A felhasználó maga bővítheti a szókincset! A felvett szavak toldalékos alakjait felismeri, elválasztja, elgépelés esetén ki is javítja!

A LEKTOR a legjobb magyar szóellenőrző, elválasztó és javító program. A közeljövőben jelenik meg a WordPerfect Windows magyar változata, mely 42 ezer szavas tezauruszt is tartalmaz.

A LEKTOR (DOS, WINDOWS) ára
12 000 Ft + áfa,
(komparatív) upgrade ára 6000 Ft + áfa

Kérjen részletes ismertetőt vagy rendelje meg

MICROSEC Kft.
212-2857, 116-8400 (Tel./Fax)
1374 Budapest, Pf. 515

A KIMSOFT IFABO ajánlata

Microsoft akció (amíg a készlet tart)

ACCESS 2.0 magyar/ Up.	**42 900,-/16 900,-**
FoxPro 2.6 Standard/Upgr.	11 900,-/ 2 496,-
FoxPro 2.6 Prof./Upgr.	**57 900,-/29 900,-**
WinWord 6.0 (magyar)	42 900,-/16 900,-
EXCEL 5.0 (magyar) / Upgr.	**42 900,-/16 900,-**
Works for Win. 3.0 (magyar)	12 900,-/ 9 400,-
Excel+WinWord+Powerpoint+ACCESS= Office 4.3 Prof.(magyar)	**61 900,-/49 900,-**
Magyar helyesirás-ellenőrző progr.	11 900,-
Windows 3.1 magyar/Up.	**13 400,-/ 9 900,-**
Win. for Workgroups Add On 3.11	6 400,-
MS DOS 6.22	**6 900,-**
MS Publisher 2.0 / Upgr.	12 900,- / 9 900,-
Visual Basic 3.0 Prof./Up.	**42 900,- /21 900,-**

Újdonságok, bevezető árak

Borland C++ 4.5 /Upgrade	49 900,-/27 900,-
Borland Delphi for Windows	**29 900,-**
CA-Visual Objects for Clipper	49 900,-
Fractal Design Painter 3.0	49 900,-
IBM OS/2 v3 Warp (IFABO akció!)	**Hívjon!**
KEDIT for Windows	21 900,-
Norton Commander 5.0	**9 990,- / 5 400,-**
Quattro Pro 6.0 Win./Upgr.	16 900,- / 8 400,-
PerfectOffice 3.0 /Upgr.	**62 900,-/31 900,-**

CD-ROM-ok, játékprogramok

MacMillan Dictionary for Children (CD)	6 400,-
TIE Fighter / Rebel Assault	**7 400,- /7 400,-**
Rise of the Robots (CD)	7 900,-
Dark Forces/Mortal Kombat II	9 600,- /9 600,-
Publishers Paradise Prof. CD	**6 900,-**
Angol-magyar szótárak	Nagy választékban!
Egyéb multimédia CD-ROM-ok	**Hívjon!**

Adobe PhotoShop 3.0	**94 900,-/37 400,-**
AutoCAD LT + magyar könyv	53 900,-
Blinker 3.1 (Linker Clipperhez)	**39 900,-**
Clarion adatbázis-kezelők	Hívjon!
Clipper+ExoSpace+Tools/dBFast	**37 900,-**
Close Up 6.0 / Upgr.	21 900,-/11 900,-
CorelDRAW 5.0 teljes magyar betűkészlet (kb. 800 db font)	**17 400,-**
CorelDRAW 5.0 CD /Up.	49 900,-/28 900,-
CorelDRAW 3.0 CD OEM ver.	**5 400,-**
Corel ArtShow 2+3+4+5	10 990,-
CorelFlow (folyamatábra-tervező)	10 990,-
Corel Ventura 5.0 CD	**39 900,-/19 900,-**
dBASE 5.0 for DOS/Win.	16 900,-/17 900,-
F-Prot 2.17 Prof. (antivirus pr.)	**Hívjon!**
Könyvelő és nyilvántartó programok	Hívjon!
QEMM 7.5 / 386 MAX 7.0	**10 900,-**
McAfee VirusScan (aktuális ver.)	17 400,-
Norton Utilities 8.0 /Up.	**18 400,-/ 6 900,-**
Novell DOS 7.0 (Akció!)	7 400,-
Novell NetWare 4.1 (5 user)	**99 900,-**
Pc Tools 2.0 for Win./9.0 DOS	15 900,-
Print Artist 2.5 for Win. CD	**8 900,-**
QuarkXPress 3.3 for Windows	97 400,-
UnInstaller 2.0 (Windows takarító)	**9 200,-**
WinCheckit 2.0 („Uninstaller"-rel)	9 900,-
WinFax Pro 4.0	**13 900,-**
Windows 3.1-hez magyar ékezetes TrueType betűcsomagok (50 db font)	4 900,-
Xtree Gold 3.0 DOS	**15 900,-**

Hardver árjegyzékünkből

SONY CDU-55E CD ROM olvasó	18 900,-
SoundBlaster hangkártyák	**Hívjon!**
HP ScanJet IIp + Recognita Select	77 400,-

A közölt árak nem tartalmazzák a 25%-os áfát, és a helyszíni üzembehelyezés költségeit.

Rendkívüli szoftverakciókkal várjuk Önöket az IFABO'95 kiállítás C pavilonjában, a 10/f standon!

KIM-SOFT Számítástechnikai és Kereskedelmi Kft.
1112 Budapest, Hegyalja út 70. fszt. 2.
Telefon: 371-5012 (fax is) és 06-30-461-058

INFORMÁCIÓKÉRÉS: 0533 ▲

PROFON

1141 Budapest, Egressy út 113/E
Tel. & Fax: 252-0663

Várjuk az IFABO '95
„C" pavilon, 10. standján!

- ⇨ Új hálózatépítő elemek
- ⇨ Új hálózatfelügyelő rendszer
- ⇨ Új elemek az energiaellátásban
- ⇨ Változatlan minőség

BIZTONSÁGOS, MEGBÍZHATÓ SZÁMÍTÁSTECHNIKAI HÁLÓZAT

INFORMÁCIÓKÉRÉS: 0549 ▲

Szerencsés csillagzat alatt dönt, ha a QWERTY számítógépet választja, mert:

Tetszőleges kiépítésben **386, 486** és **PENTIUM** számítógépek
valamint EPSON, HEWLETT PACKARD, CANON nyomtatók

RÉSZLETRE IS kaphatók!

MULTIMÉDIA, MODEMEK, tartozékok, kiegészítők, szakkönyvek
széles választékával várjuk.

QWERTY
Alapítva: 1984-ben

QWERTY High Tech KFT. – 1114 Budapest, Bartók Béla út 9.
Tel.: **166-93-77** (4 vonal), 186-88-58, T/F: 185-26-87
Nyitva: Hétfőtől péntekig 10-18 óráig



NE FELEDJE: Nevünk ott található az Ön számítógépének billentyűzetén is!

INFORMÁCIÓKÉRÉS: 0553 ▲

Újdonságokról — dióhéjban

Azon frissiben...

Kedit for Windows v1.0

Gyártó: Mansfield Software Group
Típus: Text-, illetve programozói editor
Hardverkövetelmény: Windows 3.1 futtatására alkalmas PC
Ár: 26 000 Ft + áfa

A DOS alatti KEDIT jól sikerült továbbfejlesztése. (Erről a programról részletesen szól a 21. oldalon olvasható cikk.)

Multi-Edit Professional for Windows 7.0

Gyártó: American Cybernetics
Típus: Text-, illetve programozói editor
Hardverkövetelmény: Minimálisan 386, MS Windows 3.1 futtatására alkalmas PC
Ár: 28 000 Ft + áfa

A korábbi nagy sikerű Multi-Edit Professional DOS-os változatának alaposan átdolgozott kiadása. A jól kiaknázott DDE- és DLL-támogatás folytán bárki testre szabhatja a MEW-t, belőle fordíthat, linkelhet (szerkeszthet), élhet a MAKE funkció lehetőségeivel. Más programok meglévő help rendszereit konvertálhatja át szituációérzékeny helppé. VCS (version control system) rendszerek, úgy mint PVCS, TLIB, Source Safe vagy MKS-RCS közvetlen támogatása. Jól fejlett C-szerű makrónyelv, benne struktúrák, tömbök, DLL import stb.

Ráadásnak megkapjuk a MEW teljes DLL API-ját is, amellyel kívülről, valamely magas szintű programozási nyelvvel is „rá tudunk fejleszteni" az editorra. Egyszerre max. 128 DOS, Unix vagy bináris fájl szerkeszthető. Szinkronizált interaktív File Compare! Source code tagging & browsing (nincs rá jó magyar kifejezés, de kb. annyit tesz, hogy az egymásra hivatkozó programrészek keresés szempontjából egymáshoz láncolódnak). Több fájlt átfogó globális keresés és csere. Reguláris kifejezések kezelése. Korlátlan mélységű Undo és Redo. Oszlopszervezésű blokkok, drag & drop stb.

Multimedia Cloaking v1.01

Gyártó: Helix Software Company
Típus: Memóriamenedzser
Szoftverkövetelmény: Installált XMS-manager, például: EMM386.EXE, QEMM, NETROOM, 386MAX.
Ár: 7000 Ft + áfa

Kevés a gépe alapmemóriája indulás után? Multimédiás programokat futtat? Ez a kis csomag semmi mást nem kínál, mint három olyan segédprogramot, amely felköltözik az 1 Mbájt feletti régióba, és protected módú programként ott fut. Vagyis nem a 640 kbájt és 1 Mbájt közötti területre költözik. A három driver a következő: az első a Microsoft SMARTDRV.EXE-je kiváltására alkalmas, a CD, a floppy és a harddiszk adatelérését gyorsító (cache) program, a CACHECLK. A második egy univerzális egérdriver. A harmadik a CD-olvasókhoz betöltendő MSCDEX védett módú változata. Minimum 80K megtakarítás. Ráadásként még egy memóriatallózó és szoftvertesztprogram, a DISCOVER. MS windowsos beállítóprogram áll a kényelmesebb emberek rendelkezésére.

És még korrekt módon működik is.

Simply3D

Gyártó: VisualSoftware, Inc.
Típus: Memóriamenedzser
Hardverkövetelmény: 486DX processzor, 8 Mbájt RAM, CD-olvasó, 25 Mbájt windowsos átmeneti munkaállomány (swap file), min. 800x600x256 színű videoüzemmód
Ár: 8000 Ft + áfa

Olcsó, de nagyon hatásos grafikai segédeszköz illusztrátorok, képzőművészek, grafikusok, reklámtervezők és megszállottak részére. Meglévő TrueType formátumú betűkészleteinkből varázsolhatunk elő 3 dimenziós, színes, fényezett, látványos, hivalkodó betűket. A csomagban található CD-n még kapunk 70 anyagmintát (textúrát), száznál is több 3D objektumot (motorcsónak, Porsche, gyümölcsök, gyertyák, székek, pálmafák stb...), egy multimédiás interaktív oktatóprogramot. Cégemblémák tervezésére, animációk feliratozására, egyszerű megrendelők megdöbbentésére kiválóan alkalmas.

SuperPrint 3.10

Gyártó: Zenographics
Típus: Printermanager MS Windowshoz
Hardverkövetelmény: 4 Mbájt RAM, MS Windows 3.1 enhanced módban. PostScript printerekhez nem ajánljuk.
Ár: 9600 Ft + áfa

Mindazoknak, akik unatkoznak, mialatt windowsos szoftverük nyomtat, vagy türelmetlenek, esetleg PC-s grafikáikat szeretnék minél jobban megjeleníteni nyomtatójukon, nos, ők gondolkodjanak el a SuperPrint megvételén. A szoftver nem más, mint egy 32 bites printerdriver program az egyébként 16 bites Windows alá. Támogat tetszés szerinti mátrix-, tintasugaras és lézernyomtatót. A driver képes TIFF, BMP és egyéb bitmap állományok léltrehozására is. A SuperQueue egy olyan meglepően gyors spooler, amely a nyomtatási parancs kiadását követően egy-két másodpercen belül visszaadja a vezérlést a felhasználónak, még akkor is, ha maga a nyomtatás több percre is igénybe veszi a nyomtatót. A SuperPrint beépített drivere 256 szürke árnyalattal közelíti a TrueColor, 16,7 millió színárnyalatú képet, ha viszont színes printerünk van, a SuperPrint kontrasztosítja, élesíti a képet, a színátmeneteket, azaz javítja a nyomat minőségét!

WINProbe 3.00

Gyártó: Landmark Research International
Típus: Szoftver- és hardvertesztprogram
Speciális hardverkövetelmény nincs, az MS Windows 3.x fusson
Ár: 9800 Ft + áfa

Univerzális tesztprogram. Leírni is nehéz, mennyi információ szűrhető ki a WINProbe segítségével. Ízelítő a legérdekesebb funkciókból: Windows memória- és forrásoptimalizálás, INI editor, INI clean-up, INI compare, Group File Tune (gazdátlan ikonok és ablakok felderítése), real-time status panel. *Hardvertesztek*: video, harddiszk, CD-ROM, PCMCIA, billentyűzet gombonként, hangkártya, RAM, XMS, mouse, floppy, printer, koprocesszor. IRQ, DMA, UMbite listák. *Hardver setup*: konfliktusok felderítése, feloldása. *Egyéb apróságok*: gyors fájlkeresés, duplikált állományok felkutatása, mintegy 300 tipp Windowst használóknak.

Delrina Perform for Windows v1.0

Gyártó: Delrina (Canada)
Típus: Táblázat- és űrlaptervező, illetve űrlapkitöltő program
Hardver- és szoftverkövetelmény: 4 Mbájt RAM, MS Windows 3.1
Ár: 18 000 Ft + áfa

Aki már küszködött MS Word for Windows-zal egy bonyolultabb felépítésű táblázaton, az különösen értékelni fogja a Delrina legújabb űrlaptervező programját. Drag & drop technikával, rengeteg mintapéldával (template), adatmezőkkel szinte automatikusan építi fel a legbonyolultabb táblázatot is. Gondoljunk csak egy APEH-adóbevallási ívre. Azt kb. 15 perc alatt „leprogramozhatjuk" a PERFORM-ban. Beépített matematikai kalkulációk, oszlopműveletek. Támogatott grafikus formátumok: TIFF, BMP, PCX és EPS. Árnyékolási technika, látványos határolóvonalak, hátterek, nyomógombok, menüs választási lehetőségek kitöltéskor. Wizard: pár kérdésre érdemben válaszolunk, mire a PERFORM azonnal legenerálja.

Tempora mutantur...

„Tempora mutantur et nos mutamur in illis" — szól a latin bölcsesség, amelynek a címben idézett első két szavánál szoktunk leragadni, pedig tudomásul kell vennünk, hogy nemcsak az idők változnak, hanem egyúttal mi magunk is formálódunk — olykor előnyünkre, máskor hátrányunkra. Házunk táján a kirakatba valók után böngészve, most véletlenül olyan unikumok keveredtek palettánkra, amelyek ilyen gondolatokat „rajzolhatnak" fel bennünk.

Önarckép

Hogyan változott lapunk? Milyennek ítélik a változást olvasóink? Mennyire alakult át olvasótáborunk? Javult-e lényegesen az olvasók számítástechnikai felszereltsége? Miben változott meg az olvasók igénye? — ilyen és ehhez hasonló kérdésekre szeretnénk választ kapni kétévente rendszeresen lebonyolított közvéleménykutatásunkból, melynek kérdőívét mostani számunkhoz mellékeltük. Abban bízunk, hogy mindenkinek lesz annyi „üresjárata", amelyben kitölti és postára adja azt. A kitöltött kérdőívet csak össze kell hajtani, összetűzni vagy leragasztani, és bélyeg nélkül feladható — a portóköltséget is mi vállaljuk. További „kedvcsináló" ehhez a társadalmi munkához a cégek által felajánlott sokféle nyeremény, köztük az első díj, egy 486-os számítógép, de a többi ajándék is elég vonzó.

A felmérés azért különösen fontos számunkra, mert bár folyamatosan kapjuk az olvasói visszajelzéseket, és meg is fogadjuk az így kapott ötleteket, tanácsokat, de módszeresen összesíthető, statisztikailag kiértékelhető véleményhalmazhoz csak ilyen közvéleménykutatással jutunk, és az akkor reprezentálja hitelesen az egész olvasótábort, ha nagyon sok kérdőív érkezik be. Ha tehát kérhetünk valamiben aktivitást olvasóinktól, akkor ezt a mostani közreműködést kérjük, amelynek végeredménye mindenkinek hasznos.

Lemezváltás

Lemezmellékletünk történetében a két hónappal ezelőtti szám volt a határvonal, mert lemezkapacitást és lemezmárkát is váltottunk. Az első tapasztalatok kedvezőek voltak, nem nőtt a szállítás és kézbesítés viszontagságait megsínylő lemezek aránya. A lemezmelléklet szerkesztésében a váltás nem is annyira a jóval több anyag elkészítésére fordítandó többletmunka, hanem a hozzánk beküldött (és a lemezre most már ráférő) windowsos programok rejtett aknái miatt okozott gondokat. (Lásd megjegyzésünket a DLL-fájlokról 45. oldalunkon.) De azért nem bántuk meg a váltást, örülünk, hogy így egyre több programozónak tudunk bemutatkozási lehetőséget teremteni.

Csapatösszeállítás

Változott szerkesztőségünk összetétele is. Az impresszumban Sziebig Andrea már nem szerepel, mert áprilistól „átigazolt" a CW-Számítástechnika szerkesztőségébe, annak lapjain találkozhatnak vele információszolgáltató partnereink is, olvasóink is. E havi témánk összeállítása volt Andrea „búcsúfeladata", de az általa gondozott (Unixumok, Géprajz, Mikrobazár), és a többnyire általa írt rovatok (Paletta, Böngészde, Kirakat) „visszaszálltak" arra a csapatra, amely éppen 5 évvel ezelőtt, az 1990. júniusi számmal a Mikroszámítógép Magazinból Alaplappá történő átalakulást — a lap 12 éves történetének eddigi legnagyobb fordulatát — végrehajtotta. Továbbra is megtalálhatják tehát lapunkban a termékekről, cégekről, eseményekről rövid információkat adó rovatokat is, de azokat a „kvartett" helyett egyelőre a „trió" fogja írni és szerkeszteni — legjobb tudása (és energiatartalékai) szerint.

Sajnáljuk, hogy Sziebig Andrea „elcsábult" az [Új] Alaplaptól, de azt is látjuk, hogy műfajainak igazi terepe valóban egy hetilap... És ahogy előző számunkban is kifejtettük, mi a számítástechnikai ismeretterjesztés közös ügyéért dolgozó összes kollégánkat egyetlen nagy család tagjainak tekintjük, függetlenül attól, hogy ki éppen melyik csapathoz tartozik. Tempora mutantur... de ebben nem változunk.

Faklen Pál

Előfizetés az Új Alaplapra

Az 1995/......... számtól kezdődően előfizetem az Új Alaplap című, havi számítástechnikai folyóiratot ...... példányban, ❑ 1 évre, ❑ 1/2 évre.

Az éves előfizetési díj 2 970,– forint.

Az előfizetési díj kiegyenlítéséhez:

❑ **Számlát** kérek (banki átutalással fizetek).

❑ **Átutalási postautalványt** kérek.

Név: ....

(Cég:) ....

Cím: ....

Helység: ....

Irányítószám: ....

Dátum: ....

- - - - - - - - - - - - - - - -
/aláírás/

INFORMÁCIÓKÉRÉS

Kérem, hogy az itt általam **BEKARIKÁZOTT KÓDSZÁMÚ** hirdetésekkel kapcsolatban küldjenek részemre bővebb tájékoztatást.

0501	0518	0535	0552
0502	0519	0536	0553
0503	0520	0537	0554
0504	0521	0538	0555
0505	0522	0539	0556
0506	0523	0540	0557
0507	0524	0541	0558
0508	0525	0542	0559
0509	0526	0543	0560
0510	0527	0544	0561
0511	0528	0545	0562
0512	0529	0546	0563
0513	0530	0547	0564
0514	0531	0548	0565
0515	0532	0549	0566
0516	0533	0550	
0517	0534	0551	

Beküldhető: 1995. június 30-ig

ÚJ ALAPLAP 1995/5 MÁJUS

1. Reméljük a PSION termékei felkeltették érdeklődését, és most itt az alkalom, hogy közelebbről is megismerkedjen velük. Ha válaszol kérdéseinkre és megküldi azokat számunkra, akkor posta fordultával eljuttatunk Önnek egy floppy lemezt, rajta a PSION S3a számítógép magyar nyelvű emulátor programjával.

2. A floppy lemez segítségével ismerkedjen meg a Series 3a működésével és szolgáltatásaival, ha megtetszett Önnek, válaszoljon a floppyn lévő újabb kérdéseinkre. Amennyiben a válaszokat elküldi nekünk, akkor részt vesz a jutalomsorsoláson, és nagyértékű menedzser számítógépet nyerhet.

Első forduló kérdései:

1. Számítástechnikai érdeklődési területe:
2. Használ-e PC-t? (i/n):
3. Milyen célra?:
4. Igénybe vesz-e nemzetközi hálózati szolgáltatást?
   Compuserve: Internet: Egyéb:
5. Használ-e GSM mobil telefont? (i/n): Milyen típust?:
6. Használ-e kézi zsebszámítógépet? (i/n): Milyen típust?:
7. Ha rendelkezik PSION kéziszámítógéppel, elégedett-e a szolgáltatásaival?:
8. Milyen bővítést, kiegészítést igényelne?:
9. Tervezi-e, hogy a közeljövőben kéziszámítógépet vásárol?:
10. Mire kívánná használni?:

PSION

FELADÓ:

Név:

Foglalkozás:

Cég:

Cím:

Főnyeremény: PSION Series 3a 512 KB magyar nyelvű hordozható menedzser számítógép (95000,- Ft)

II. díj: S3a 256 KB magyar nyelvű hordozható számítógép (85000,- Ft)

III. díj: S3 256 KB angol nyelvű hordozható számítógép (65000,- Ft)

Itt hajtsa össze!

PSION
Magyarország Kft
Budapest
Csörsz u. 3-5.
1123

Bélyeg helye

Bélyeg helye

Új Alaplap
szerkesztősége
Pf. 571
Budapest
1538

Feladáskor kérjük bérmentesíteni!

FELADÓ

A) Egyéni érdeklődő:

Név:

Cím:

Helység:

Irányítószám:

B) Vállalati érdeklődő:

Cég:

Ügyintéző:

Cím:

Helység:

Irányítószám:

Telefon/Fax:

Belföldön díjmentesen feladható

Új Alaplap
szerkesztősége
Pf. 571
Budapest
1538

Minden PC-hez kell egy jó alaplap!

És egy Új Alaplap!

Az Új Alaplap lemezmellékletének tartalma **1995 / MÁJUS**

- ❑ Egy Awk-változat — AWK#.EXE [Cikk a 14. oldalon]
- ❑ Kiegészítés az előző szám(ok) windowsos programjaihoz — DLL#.EXE
- ❑ A Multikey programról — MULTIKEY.TXT, MULTIKEY.COM, MULTIKEY.DRV
- ❑ A RAR tömörítő shareware változata — RAR.TXT, RAR#.EXE [Cikk a 45. oldalon]
- ❑ Egy Dbase-kompatibilis adatbáziskezelő — DBKER.DOC, DBKER#.EXE (Boncz István Tamás)
- ❑ Karaktersorozatok cseréje — MUHELY#.EXE (Lois László) [Cikk a 49. oldalon]
- ❑ Postai adatok adatbázisa — POST.TXT, POST#.EXE (Oláh Gábor)
- ❑ Matematika alapműveletek gyakoroltatása — MATALAP.TXT, MATALAP.EXE (Simay Endre István)
- ❑ Háromdimenziós játék: David's Wordy — DWORDY.TXT, DW#.EXE (Dávid Norbert)



DEC: nagysebességű LAN-adapterek

A Digital új hálózati csatolókártyát fejlesztett ki személyi számítógépeire. A Fast EtherWORKS PCI 10/100 LAN-adapter kis CPU-teljesítménnyel gyors adatátvitelre, és mind 10, mind pedig 100 Mbit/s-os átviteli sebességre képes. Fejlesztésekor elsősorban azokra a ma 10 Mbit/s-os hálózatot használókra gondoltak, akik a jövőben a 100 Mbit/s-osra kívánnak átállni. A kártya a tengerentúlon 249 USD-be kerül, és május-júniusban jelenik meg a piacon. Ide tartozik még az a hír, hogy az EtherWORKS Turbo PCI adaptereinek maximálisan 28%-os árcsökkentésével a Digital a 10 Mbit/s-os Ethernet PCI kártyák nemzetközi piacán az ár/teljesítmény arány szerint az élre került.

Xircom: mobil hálózatok szakértője

A Walton Kft. március végi Multi Vendor Day rendezvényén kilenc vezető nemzetközi cég között bemutatkozott a Xircom is. Ez a vállalat korán felismerte, hogy a hordozható számítógépek elterjedésével fokozódik az igény a mobil, kis energiaigényű LAN-csatlakoztatást biztosító eszközök iránt. Ezek a kívánalmak vezettek a díjnyertes Xircom adapterek kifejlesztéséhez. A legszélesebb körű kompatibilitás érdekében egyre több driverrel látják el ezeket a termékeket, így összekapcsolhatók a legelterjedtebb desktop hálózati operációs rendszerek legújabb verzióival.

A Xircom termékeknél biztosított az NDIS- és az ODI-támogatás csakúgy, mint a pocket driver TCP/IP-hez. A jövőben a cég főként a vezetékes és a vezeték nélküli hálózatok közötti kapcsolatok gördülékeny üzemeltetésének feltételeit igyekszik elősegíteni, beleértve a hálózatok távoli elérhetőségét. A Pocket és a CreditCard adapterek segítségével hordozható PC-kkel csatlakozhatunk vezetékesen a hálózathoz, a szintén a cég termékének számító Print Serverrel pedig lehetővé teszik a rendszermérnököknek, hogy a felhasználók szempontjából a legalkalmasabb pontokon helyezhessék el a nyomtatókat. A Xircom legfrissebb fejlesztései a vezeték nélküli NetWave termékek. Segítségükkel lehetővé válik hordozható gépek vezetékes hálózathoz való kapcsolása vezeték nélkül bárhonnan, működés közben akár még mozgatható is a felhasználói termék — pl. épületek között — anélkül, hogy a kapcsolat megszakadna.

Megjelent a cég kínálatában a CEM2 típusjelű termék, amely lokális Ethernet és távoli fax vagy modem kapcsolatát valósítja meg egyetlen kártyában. Időközben a NetWave termékeket sikeresen tesztelték az ETSI laboratóriumában, ETS (European Telecommunications Standard) 300328 szám alatt megkapták a szabványosság „bizonyítványát", így az európai piacon is forgalmazhatók.

NetWorth: a NetWare-barát

A NeWorth cég fejlesztései főként a nagy sebességű hálózati és kapcsoló hub technológiákra irányulnak. Kiemelt fontosságú a Fast Ethernet (100BASE-TX) technológia, amelynek megfelelő termékeket a NetWorth az elsők között hozott forgalomba igen kedvező áron. A már említett Walton-rendezvényen a céget képviselő szakember ismertette, miként növelhető a hálózat sávszélessége szegmentálással, kapcsolástechnikával (switching) és Fast Ethernet alkalmazásával. Ehhez kapcsolódóan kitért a forgalmi zavarok csökkentésére (sokszegmensű hubok), a szerver-hozzáférésnek mint szűk keresztmetszetnek a kiküszöbölésére (kapcsolt Ethernet, nagysebességű szerver-hozzáférés).

A téma különösen érdekes NetWare hálózatok esetén, amelyekre a NetWorth speciális, terhelést optimalizáló NLM-készletet kínál. A cég egyébként az ún. stackelhető hub rendszerek terén 10%-os részesedésével negyedik a világpiacon. Legújabb ilyen terméke a FastStack, az első teljesen integrált stackelhető hubrendszer, amely kombinálja az Ethernet kapcsolási technikát, a 100BASE-T hubokat és a gazdaságos 10BASE-T hubokat. A Fast Stack alapja a cég által kifejlesztett FastPipes kapcsolástechnológia.

Retix: a „switchmester"

Az amerikai Retix hálózati eszközgyártó cég számára kiemelt jelentőségű a fejlesztés alatt álló új termék, az Enterprise NetWork Switch (ENS), amely egy egységben tartalmaz multiprotokollos routert és Ethernet switch-et. Az ENS támogatja a Fast Ethernet FDDI- és ATM-szabványokat, valamint virtuális LAN kiépítését. Az eszközökbe szükség esetén WAN-portok is beilleszthetők. A Retix (kétirányú) gyűrűvel stackelhető nagysebességű UTP switch eszközeivel több épületet átfogó, osztott gerinchálózatok alakíthatók ki. Az új termékek sorában találjuk a MetroLAN 5000 típusjelűt, amely az első osztott Ethernet switch városméretre kiterjedő hálózati szolgáltatásokra tervezve. Továbbá: a Retix bejelentette a NETXchange 4000 típusjelű, olcsó Ethernet bridge/router termékcsaládot, amelyet leginkább költségérzékeny hálózatok kialakításánál javasolnak figyelembe venni.

Lemonet: UB Networks-képviselő

Az amerikai UB Networks cég az ún. Enterprise networking vezető fejlesztője és gyártója. A budapesti Lemonet Kft. az UB Networks és más neves gyártók (ACC, Cisco, Wellfleet, Novell, AT&T) termékeinek felhasználásával intelligens információs hálózatok építőelemeit forgalmazza. A két éve alakult kft. azért választotta az UB Networks céget (amelynek egyébként disztribútora), mert az a technológiák széles választékát nyújtja; vezető szerepe van a hálózati világban, és több mint száz országban 20 ezer cég bizalmát élvezi.

Az UB Networks a Las Vegasban rendezett Networld + Interop kiállításon jelentett be InterWeb és InterWorld VR néven két új terméket. Ezek segítségével interaktív 3D-s, többfelhasználós üzleti alkalmazások fejleszthetők, és az Internet világhálózaton, továbbá más szélessávú privát adathálózatokon lehet őket alkalmazni.

IBM: hálózati szoftverözön

A legújabb IBM Networking c. katalógus mintegy 150 hálózati terméke közül 20 teljesen újnak számít. Ilyen a hardvereszközök terén az EISA Ethernet Adapter, ami egy 32 bites „busmaster" csatoló; az IBM Wireless LAN, amely vezeték nélkül LAN állomások összeköttetését oldja meg; új FDDI adapterkártyák, amelyek mindegyike kompatibilis a vezető operációs rendszerekkel.

A szerverszoftverek között az IBM LAN Server for AIX 1.1 megnyitja a világot az AIX előtt a PC-bázisú kliensek felé. A kommunikációs csomagok között most találkozunk először az IBM Desktop SNA for AIX V. 1.1-gyel, amellyel desktop környezetet lehet SNA összekapcsolhatóságúra felerősíteni.

A nyílt rendszerek felé vezető utat egyengeti két másik vadonatúj kommunikációs programtermék: az IBM SNA Application Access for AIX V. 1.1 és az IBM SNA Client Access for AIX V. 1.1. A Windows-alapú PC-k osztott hálózati megoldásaira írták az APPC Networking Services for Windows szoftvert. AS/400-alkalmazásokat lehet Windows-alapú PC-kre áttenni az IBM Personal Communications AS/400 V. 4.0 for Windows nevű programcsomaggal. A multiprotokollos hálózati megoldások hajlékony megvalósítását segíti az AnyNet SNA Over TCP/IP Gateway for OS/2 kommunikáció-vezérlő szoftver, amely megengedi az SNA-alkalmazásoknak, hogy minden további nélkül TCP/IP hálózatokon használjuk őket.

Végül az IBM LAN Network Manager for AIX segítségével a multiprotokollos RISC System/6000-alapú lokális hálózatok minden erőforrása könnyen menedzselhető ugyanarról a munkaállomás-képernyőről és ugyanazokkal a könnyen használható szerszámszoftverekkel.

Kovács Attila

Ha lesz...

Ha egyszer végre megszületik a Windows 95, ott lesz a DEC Plug and Play asztali számítógépein és notebookjain — ezt rögzíti a Digital Equipment és a Microsoft megállapodása.

Egyébként májusban a kereskedők és a legnagyobb felhasználók számára egy Windows 95 Preview névre hallgató csomagot bocsátanak ki néhány ezer forintos önköltségi áron. A csomag a végső bétaverziót tartalmazza CD-n és floppyn, a szokásos dokumentációval együtt. Célja, hogy a szakmabeliek és főként a Microsoft partnerei még a hivatalos verzió forgalomba kerülése — várhatóan szeptember eleje, ill. a honosított magyar változatnál az év vége — előtt megismerjék az új operációs rendszert. Ezzel megelőzhető, hogy a program megjelenésekor a kereskedők csak annyit tudjanak a szoftverről, mint az első vásárlók. Gondoltak a szoftverkalózokra is, hiszen adódik az ötlet az olcsó Preview további használatára, ill. lemásolására. A Preview program a hivatalos változat tervezett megjelenése után csupán néhány hónapig — várhatóan az idén — működik. Utána csak a DOS felület használható, a grafikus interfész nem indul el.

Elvarázsolva

Tradicionális kapcsolat fűzi egymáshoz — nem csak Magyarországon — a DEC-et és a Magic Software Enterprise-t. Ez a kapcsolat most tovább mélyül: a DEC saját belső rendszereinek fejlesztéséhez is a Magicet választotta, a Magic Windows NT-s változatát pedig közösen fejlesztik a Digitallel. A Budapesten tartott közös szemináriumon bemutatták a Magic Alpha OSF/1-es változatát.

Árvihar

Egy régi igény kielégítésének szándékával okoz minden bizonnyal árvihart az Ethernet-kártyák piacán a maga 2929 forintos viszonteladói árával a Crown-Tech forgalmazta Tornado. A cél: a kevesebb pénzzel bíró felhasználók kiszolgálása a D-Link csatolóknál már megszokott minőségben. A hálózati eszközöknél eddig nem alkalmazott megoldást választva nagy értékű OEM-szerződést kötött beszállítójával a Crown-Tech — nevüket, tapasztalataikat és fejlesztési tanácsaikat adva a termékhez.

A csatolókat jelenleg ISA-buszon, BNC- és UTP-csatlakozókkal szállítják, magyar nyelvű leírással, és driver szoftverrel az összes jelentős hálózathoz. A Tornado-termékek minőségét az ISO-9001 és a Novell Tested & Approved minősítés tanúsítja.

Az Intel ad lendületet a SZÜV-nek

Az állami vállalat, majd állami rt., s napjainkra többségi magántulajdonú SZÜV Rt. tevékenységének megújítását többek között az Intelhez fűződő szorosabb kapcsolattól reméli. A Computer-M inkább csak mint szaküzletek hálózata volt ismeretes szerte az országban, várhatóan azonban mint CM számítógép-előtag is népszerűségre tehet szert. Erre jogosít fel az „Intel inside” felirat: a CM Classic, CM Performance és CM Server családok mindegyikében nemcsak a mikroprocesszor származik a világ legjelentősebb félvezetőgyártójától, hanem az alaplapok is. A nevükkel jelzett teljesítménykategóriáknak megfelelően az árakban is igen nagy a szórás: a „klasszikus” 486-SX 33-as CM 118 900 forintba kerül, a CM Server MX (Pentium 90-essel, 16 Mbájt RAM-mal, 1 Gbájtos SCSI winchesterrel) nettó listaára 847 800 forint. A gépekhez 2, illetve 3 év garancia jár (sajnos a monitorokért külön kell még fizetni: 15 900 forinttól 312 000 forintig terjed az áruk).

Ingyen Psion!

No nem a népszerű Series 3a, csak annak egy teljes értékű, PC-n futó bemutatószoftver-változata. A téma iránt érdeklődőknek csak egy kérdőívet kell kivágniuk lapunk lemezvédő kartonjából, és máris még egy — szakmailag ugyancsak tanulságos — lemezmellékletet „kasszírozhatnak” egy Új Alaplap árán.

Évfordulók

Az Allegro Bt és a Flag Kft is júniusban ünnepli ötödik születésnapját, az Albacomp pedig éppen 10 esztendős. Az évfordulót mindenki más "hangszerelésben" ünnepli. Az Albacomp egy hagyományos rendezvény keretében „szerényen” bejelentette, hogy országos szervizszolgálata és viszonteladói hálózata mellé kiépíti szaküzleteinek országos láncolatát is.

Az Allegro egy sor csemegeújdonsággal szolgál a műfaj barátainak: ilyen a Fast Movie Machine II (overlay, tv-tuner, home studio), a Fast FPS 60 vagy a Fast PCI Video (2 MB PCI). Opcionálisan mindháromhoz elérhető a Motion-JPEG 50/60 két félképes rögzítéssel. A PC Audio/Video utómunka rendszerekként üzembe helyezett Fast Video Machine termékcsaládból eddig 54 talált gazdára Magyarországon (12 újabb előkészítés alatt), és egyre népszerűbbek a hibrid rendszerek, amelyek mind a hagyományos, szalagos utómunkát, mind a merevlemezen történő szerkesztést lehetővé teszik. S hogy a kereslet továbbra sem lankad, jól mutatja, hogy az Allegro magyarországi eladásai a Video Machine rendszerek területén meghaladják a lakosságában összemérhető, de gazdasági lehetőségeiben messze a magyar felett álló országok némelyikében értékesített VM rendszerek számát (például Svédországét).

A Flag Kft alapítói öt évvel ezelőtt azzal számoltak, hogy a gyorsan fejlődő magyar számítástechnikai piac igényli a kereskedelemtől a rendszerszervezésen át a szervizt és a karbantartást is magába foglaló komplex szolgáltatást. A cég eredményei azt igazolták, hogy jól mérték fel a várható piaci fejleményeket: tavalyi forgalmuk 20-szorosa volt az indulás évében regisztráltnak. Ehhez köze van persze a jó partnerválasztásnak is: az OKI, a Compaq, az Intel, a Novell, az AT&T vagy a Microsoft is odatette a maga „ajándékát” a születésnapi torta mellé, amelyet igen ötletes műsoros ünnepségen, „keleti pompával”, a Gellért Fürdő csarnokában szolgáltak fel szakmai partnereiknek.

Grafikus munkaállomás

Április közepétől szállítja az ALR a jelenleg legnagyobb teljesítményű, 120 MHz-es Pentiummal működő grafikus munkaállomását. A gép alapkiépítésben 16 MB RAM-ot, 256 cache-t, PCI lokálbuszt, Adaptec SCSI vezérlőt, 1,44 MB-os floppy-, és négyszeres sebességű CD-meghajtót tartalmaz. Merevlemeze 2 GB-os SCSI, előre telepített MS DOS-szal és Windows for Workgroupsszal kerül forgalomba. Az előrejelzések szerint nyárra már a 133 MHz-es Pentium processzoros ALR gépek is várhatók.

Levelez a hivatal

Mégpedig az FTP TCP/IP protokollját felhasználva. Az FTP szoftverek hazai disztribútorával, az Areco Systems Kft-vel kötött megállapodás keretében a Miniszterelnöki Hivatal kedvezményes feltételekkel jut hozzá azokhoz a kommunikációs eszközökhöz, például az OnNet-hez, amelyek révén gyakorlatilag akár egy notebookról is elérhetővé válik a nagyvilág. Az információs adatországúton a nagyvilágot e pillanatban az Internet reprezentálja, az FTP szoftvercsaládja pedig a minél kényelmesebb Internet-hozzáférést igyekszik elősegíteni. Lelki szemeinkkel már látjuk, milyen jó boltot csinál majd egy hozzáférés, illetve zaklatás elleni védelemben utazó fejlesztő cég, ha körül kell bástyáznia a hivatalt szűkebb pátriájának „áldó imádságaival” szemben.

Pályázat

A Creative Engineering Kft. a Parametric Technology Corporation magyarországi disztribútora, a Pro/Engineer 3D-s CAD/CAM rendszer forgalmazója pályázatot hirdet egyetemi és főiskolai hallgatók részére a tervezéstől a gyártásig a Pro/Engineer rendszerrel témában.

A pályázat tárgya a Pro/Engineer rendszer felhasználásával készült munka (diploma- vagy TDK-munka, vagy ezen munkák része). A pályázatnak tükröznie kell a Pro/Engineer lehetőségeit, az alaksajátosságok szerepét a mérnöki munkában a koncepcionális tervezéstől a gyártásig, a Pro/Engineer szakmoduljainak lehetőségeit.

A pályamunkák beadási határideje: 1995. június 30., a pályázatokat a Creative Engineering Kft. munkatársai bírálják el, és a nyertesek között az alábbi díjakat osztják ki: 1. díj — 40 000,– Ft, 2. díj — 20 000,– Ft, 3. díj — 10 000,– Ft.

Információ, szakmai segítség Kőrösi Árpádtól, illetve Szabó Józseftől (Creative) kérhető.

Ötéves az Ifabo

A szakma terített asztala

Az ötéves Ifabo öt Európa-csillaggal jelölte meg a számítástechnikát, a telekommunikációt, az irodaszervezést, a másolástechnikát és az irodabútorokat — mint az idei kiállítás öt fő témakörét, összetevőit a nemes konyak aromájának. Valószínűleg azonban az ünnepi koccintásra emelt poharakba kerül némi fanyar íz is. Van olyan érzésünk, hogy a számítástechnika idén először lesz kénytelen eljátszani a „szegény rokon" szerepét az ünnepien megterített asztalnál. Az alacsony árrés-szintre berendezkedett szakmának ugyanis fogyóban vannak a tartalékai, a jelenlegi gazdasági körülmények nem igazán kedveznek neki, a konjunktúrának pedig az istennek sem akaródzik eljönnie. De azért látnivaló most is lesz bőven.

Datateq

A Datateq Kft indította el azt az új, alapvetően magyar felhasználóknak szánt, magyar fejlesztésű online szolgáltatáscsomagot, amelynek a szélesebb nyilvánosság előtti bemutatkozása szintén az Ifabón lesz. A rendszer egy open VMS operációs rendszerű DEC 3000-es szerverre épül. A rendszer jelenleg 8 (budapesti) kapcsolt telefonvonalon érhető el, rövidesen megvalósul az X.25-ös bővítés is.

Bárki, aki csatlakozik a rendszerhez, a nap 24 órájában bármikor, saját számítógépével egy telefonvonalon keresztül csatlakozhat a Datateq központi számítógépéhez, onnan adatokat, információkat hívhat le, számítógépes szolgáltatásokat vehet igénybe.

Kezdetben inkább csak vállalkozások, cégek számára értékes információkat tartalmaz a rendszer, illetve általuk jól hasznosítható szolgáltatáscsomaggal rendelkezik. A későbbiekben közhasznú információk (országos kulturális ajánlat, turisztikai információk, menetrendek stb.) révén szélesebb lakossági csoportok számára is vonzóvá válhat a rendszer.

A jelenlegi szolgáltatáskészletnek része egy termékbörze, ennek adatait a rendszerhez csatlakozott cégek adják, s akár naponta többször is módosíthatják kínálatukat. Minden cégnek külön „boltja" van, ezen belül „osztályokat" hozhat létre, s ha egy vevőnek egy termék megtetszik a „kirakatban", e-mailen azonnal meg is rendelheti. A Datateq rendszere megkülönböztet az e-mailen belül belső, illetve külső levelezést. A belső értelemszerűen a Datateq-felhasználók egymás közti levelezését — elektronikus postaládaszerű üzenettovábbítás — szolgálja, a külső pedig az Internet-kapcsolatot jelenti. Működő szolgáltatás a telekonferencia — amit valaki begépel, azonnal megjelenik minden résztvevő képernyőjén —, valamint a fájltranszfer, amelynek keretében bármilyen formátumú fájl átküldhető vagy az összes, vagy pedig a néhány kiválasztott felhasználónak.

A rendszerben jelenleg elérhető adatbázisok között olyanok találhatók, mint: csőd-, felszámolási adatok; nemzetközi kereskedelmi adatok (keres-kínál, partnert keres, tenderek); magyar kereskedelmi adatok; magyar céginformációk. (Elérési útvonal: N81 153-4333, jelszó: demo1, demo2, demo3.)

Ericsson

Az elmúlt évek során az Ericsson kifejlesztette a szinkron digitális hierarchia (SDH) építőelemeit, amelyek már rendelkezésre állnak a magyar piacon is. Az SDH technika a PDH (pleziokron digitális hierarchia) rendszereit többek között abban szárnyalja túl, hogy az átviteli sebesség a PDH-s 2, 8, 34 és 140 Mbit/s értékeket 155, 622 Mbit/s, illetve 2,5 Gbit/s „fokozatokkal" váltja ki. E technika nagysebességű optikai/elektromos végberendezéseire, valamint a betápláló/leágaztató (Add/Drop) multiplexerekre az Ericsson már megszerezte a Hírközlési Főfelügyelet típusengedélyét, amely a szinkron digitális rendszerelemek forgalomba hozatalára és rendszeresítésére jogosítja fel.

Az Ifabón — mint Ericsson-újdonság, a fentieken kívül — a mikrohullámú átviteltechnikai berendezések MINI LINK rádiórelé-családjából az ML 23-C kelthet figyelmet. A közepes kapacitású mikrohullámú eszközcsalád rendelkezik beépített intelligens felügyeleti rendszerrel is. Egy mikroprocesszoros „őr" figyeli a funkcionális hibajelzéseket, egy adatbuszra juttatja azokat, ahonnan például PC segítségével a hálózat bármely pontjáról lekérdezhető az információ. Így a szervizmérnök gyors áttekintést kap a hálózat állapotáról, további közeli és távoli hurkok ki- és bekapcsolásával tesztelheti a rendszert. (Van egy független szervizcsatorna is.)

A nyilvános és magánhálózatokban a 10–50 km-es pont–pont közötti összeköttetések megvalósítására (max 240 PCM — impulzus-kód modulált — hang- vagy adatcsatorna átvitelére) alkalmas MINI LINK működési frekvenciatartománya 15–35 GHz, de az ún. csökkentett sávszélességű változatban illeszkedik a 3,5; 7; 14 MHz-es frekvenciaraszterhez. Elsősorban celluláris hálózatokban (GSM) a bázisállomások és a központok összekötésére, de számítógépes adatátvitelre is kiváló, nagy megbízhatóságú (ISO 9001 ajánlásainak megfelelő), könnyen telepíthető, kedvező árszintű rendszer(család).

Lias

A Lias Kft mint hálózati integrátor arra törekszik, hogy optimális megoldást kínáljon hang-, adat- és videohálózati szolgáltatást igénylő ügyfeleinek. Az Ifabón a Lias bemutatja, hogyan illeszkednek egymáshoz a legkülönbözőbb informatikai alkalmazások a korszerű infrastruktúra felhasználása révén: egyetlen kábelhálózaton mutatkozik be telefon, kábel-tv, 100 Mbit/s-os adathálózat és 150 Mbit/s-os ATM. Középponti helyre kerül a kiállításon az AT&T Systimax strukturált kábelezési rendszer, amelynek a Lias értéknövelő viszonteladója, de ott található a (részben) kábelek nélküli megoldás, a szórt spektrumú rádiós technológián alapuló WaveLAN is. A képviselt cégek sorában ott lesz még a Cabletron, a HP, az SMC, a RAD és az AVM is.

Microsoft

Bármennyire meglepőnek tűnik, a Microsoft magyarországi képviselete saját standdal első ízben az idei Ifabón vesz részt. A Microsoft komolyan veszi a magyar piacot (is), már tizenhárom magyarított programot tudhat magáénak a sokezernyi felhasználó, ezek legfontosabbjai, és a köréjük épített szolgáltatások mutatkoznak be az Ifabón.

S bár stratégiailag a legszélesebb alkalmazói kört tekintik a sajátjuknak, ezúttal inkább az üzletemberekre és vállalati döntéshozókra koncentrálnak: az Office, a Works, a Windows NT-re épülő hálózati alkalmazáscsomag, a BackOffice elsősorban az ő igényeiket hivatott kielégíteni.

Lehet, hogy idei dátumúként utoljára itt próbálhatják ki az érdeklődők a megjelenésében bizonytalan — bár az augusztusi (?) világbemutató után év végéig már magyar változatban is megígért — Windows '95 legutolsó bétáját.

Természetesen ott lesznek a Microsoft-standon a magyar fejlesztőpartnerek: látható lesz például a Microsoft Mailre épülő, már működő országos elektronikus levelezőhálózat, és több más hazai készítésű alkalmazás is.

Kiköltözik a vásárvárosba a cég két alapvető telefonos szolgáltatása: a szoftverinformáció és a forródrót, tehát személyesen is lehet érdeklődni.

Rich Selling

Az elmúlt Compfairen debütált Rich Selling szoftver sokoldalú alkalmazhatóságának és kedvezményes bevezető árának köszönhetően igen kedvező visszhangra talált. Most a direkt értékesítésen túl a regisztrációs értékesítési módszer bevezetését határozták el. A cég piacelemzéséből ugyanis az derült ki, hogy azok a — kezdő — felhasználók, akik a PC-vel együtt komplex felhasználói rendszert kapnak, könnyebben és kevesebb kudarc árán barátkoznak meg feladataik — olykor kényszerű — számítógépi megoldásával. (Az elsősorban érintett kör itt az a kisvállalkozó, aki napi ügyvitelét kénytelen számítógépen nyilvántartani.) Ezek a vásárlók később is szívesen fordulnak azokhoz a forgalmazókhoz, akik már a kezdés pillanatában kész megoldásokat kínálnak számukra. A Rich programcsalád alapváltozatát díjmentesen kapják meg a partnerek, s azt korlátozás és vásárlási kötelezettség nélkül másolhatják, terjeszthetik — használhatja bárki. Ha viszont valaki a programcsalád bármely tagját egyedi igényei szerint kívánja használni, jogot már csak a regisztráció (20 000 forint) révén szerezhet rá. A Rich programcsaláddal a SZÜV standján ismerkedhetnek az Ifabo látogatói.

SPSS

Eljön az idei bemutatóra az SPSS is, amelynek — az angol Windows Magazine szerint — bármely más professzionális statisztikai csomagnál könnyebb a használata. (Statisztikai programnál ez igencsak nagy dicséret.) Igaz volt ez már a korábbi, más gépes, más operációs rendszerre írt inkarnációkra is, és — állítólag — érdemes volt kivárni a kissé hosszúra nyúlt windowsos fejlesztés végét: méltó elődeihez. Az SPSS bejövő információi készülhetnek Excellel, Lotusszal, Dbase-zel, de képes fogadni SQL serverek vagy Oracle adatbázisok anyagait is. A nagy számításigényű statisztikai programoknál nem elhanyagolható az időtényező: egy közepes bonyolultságú mátrixon az SPSS for Windows lényegében a DOS-os változatnak megfelelő idő alatt végzi el a számításokat. A mindössze 1,5 Mbájtnyi helyet foglaló program elméletileg egy 286-os gépen is elindul, de a javasolt minimum inkább egy 4 MB-nyi RAM-mal bíró 386-os konfiguráció.

Sun

Sajátságos „kompetitív upgrade"-re szólít fel a Sun hazai disztribútora: más gyártók „megunt" munkaállomásairól való áttérés esetén mintegy 6000 dollárnyi kedvezménnyel vásárolhatnak a Sun 20-as gépcsaládjának legnagyobb teljesítményű tagját, a SPARCstation 20/61-est választók. Ugyanakkor az SS20/50-esek saját upgrade-je révén pedig 12 000 dollár takarítható meg.

Sagax

Végül egy olyan eseményről, amelyre nem az Ifabo helyszínén, de az Ifabo idejében kerül sor: május 12-én, a Műegyetemen — csakúgy, mint az elmúlt év őszén — tartja ingyenes szakmai bemutatóját, a 3. Electronic Design Show-t a Sagax Kft. A nyomtatottáramkör-tervező szoftverek piacán jelentős Advanced PCB (2.5-ös változat) mellett megismerkedhetnek a résztvevők a Protel Technology Advanced Schematic 2 nevű termékével, valamint a Microsim Corporationnek és a HEM Datának a témához illő szoftvereivel.

Jakab Ágnes—Varga János

E számunk hirdetői

Cég	Info#	Oldal
Aerus	0501	34.
Aktív Rekord	0502	40.
Areco	0503	02.
AT&T	0504	B3.
Budapest Papír	0505	25.
CBI	0507	57.
C.Computer	0506	20.
Cégszerviz	0515	19.
Compmark	0508	44.
CompuServe	0509	50.
ComputerBooks	0510	57.
Controll-Szeged	0511	K4.
Corwell	0512	24.
Creative Engineering	0513	53.
Crown-Tech	0514	55.
DIT Computer	0516	20.
Electraplan	0517	57.
Elender	0518	53.
Envicom	0519	48.
Eper Stúdió	0520	44.
Fefo	0521	29.
Foxtrend	0522	18.
Gemma	0523	24.
Gemofis	0524	02.
Grafológiai Intézet	0525	47.
Halaspack	0526	23.
Hód	0530	20.
Holland Rt	0527	18.
HumanSoft	0528	34.
Hunix	0529	17.
Intergraph	0531	62.
Keszo	0532	K4.
Kim-Soft	0533	30.
Landinfo	0534	22.
LSI Oktatóközpont	0535	48.
Microsec	0537	30.
Microsoft	0538	43.
Műszaki Könyvkiadó	0539	29.
N-SYS	0540	23.
NetStar	0541	20.
Novell	0536	B2.
NYÁK Iroda	0542	53.
OKI	0543	44.
Onyx	0544	21.
OpenSoft	0545	29.
Oracle	0546	B4.
PC Kuckó (Digitrade)	0547	29.
Procomp	0548	34.
Profon	0549	30.
Psion	0550	K1.
Pákász	0551	K4.
Quantum	0552	20.
Qwerty	0553	30.
RezonTrade	0554	57.
Rich Selling	0555	53.
Sagax	0556	30.
ScanDer	0557	34.
Schwar	0558	19.
SCI Modem	0559	40.
Spieler	0560	24.
Szentendrei Papírgyár	0561	B3.
TeleLogic	0562	19.
Teta	0563	02.
Time-Life	0564	Melléklet
Walton	0565	23.
Zeller (Média Comp.)	0566	19.

Csökkenthető a bonyolultság?

Nincs kockázat, van RISC!

A RISC is azon „varázsszavak” egyike, amelyek az utóbbi néhány évben egyre többször hangzanak el a számítástechnikával kapcsolatban. A szakmabeliek tudják, hogy processzorról, pontosabban processzorokról van szó, de pontos jelentését kevesen ismerik. A RISC processzorok az utóbbi években egyre inkább felváltják a korábbi CISC típusokat, például az Apple a Motorola 68xxx sorozat helyett a PowerPC CPU-kat alkalmazza új gépeiben. A CISC—RISC kategorizálás egyre inkább túlhaladott, a CISC processzorok vagy kihalnak — mint az említett Motorola-sorozat —, vagy amennyire csak lehetséges, alkalmazzák a RISC módszereket (mint az Intel 486 és a Pentium).

A CISC-architektúra gyökerei visszanyúlnak a 60-as évek közepéig, az IBM 360-as és 370-es sorozatához; ezek az első kiforrott, széles teljesítménytartományban gyártott rendszereknek tekinthetők. A PC-korszakban tipikus képviselői ennek az architektúrának az Intel 80286/80386, a Motorola 68020 vagy a VAX-11/780.

Legfontosabb jellemzője, hogy a fejlesztés során két célt tartottak egyszerre szem előtt: a minél gyorsabb programvégrehajtás mellett azonos fontosságú volt a magasszintű nyelvek fokozott támogatása. Ezt mutatta az utasításkészlet folyamatos bővülése egyre bonyolultabb utasításokkal (mint az egész számok szorzása vagy osztása, a memóriaelérés, vagyis a címzési módok növekedése, valamint a használható számformátumok — egész és lebegőpontos — növekedése).

A chipgyártás technológiája a processzorok sebességének sok paraméterét eleve meghatározza. Ilyenek például: a használható maximális órajel, a hőtermelés, az elemsűrűség, de az adott költségszint mellett telepített tranzisztorok száma is. Ezen túl a teljesítmény már csak az utasításdekódolás és -végrehajtás hatékonyabb megoldásával fokozható.

Ezért terjedt el a korábbi cikkeinkben már említett utasításcső (pipeline vagy egyszerűen csak pipe, azaz cső), amely egy futószalagnak tekinthető. Az utasítások az egyenkénti dekódolás és végrehajtás helyett ezen a több „szerelőponttal” — fokozattal — ellátott futószalagon haladnak végig. A szokásos fokozatok az utasításlehívás, -dekódolás (memóriacím kiszámítása, végrehajtás, memóriába írás).

Ne kelljen „csőgörény”!

A végrehajtó fokozatot gyakran két részre bontják. A módszer legnagyobb előnye, hogy a CPU így egyszerre, egy időben több utasításon dolgozik, az említett 4 fokozatú csőben négy utasítás lehet bent. Ha minden utasítás csak egy órajelnyi időt tölt az egyes fokozatokban, akkor a cső elején minden órajelre betölthetünk egy újat. Ekkor egy adott utasítás végrehajtása a négy fokozaton végighaladva négy órajelet igényel, a CPU teljesítménye azonban egy utasítás minden egyes órajelre.

Ehhez persze alapfeltétel, hogy az utasítások a csőben fennakadás nélkül haladjanak végig, és ezen a ponton a fejlesztők szembekerültek a CISC rendszerek bonyolult, a magasszintű nyelveket támogató utasításaival. Nézzünk erre egy példát. Az Intel x86 CPU-k ismerik az *ADD konstans, memóriacím* utasítást, amellyel hozzáadhatunk egy konstans számot a memóriában tárolt értékhez. A programozó, illetve a fordítóprogram ezzel az utasítással a memóriát egyszerűen egy számtömbnek tekintheti. A CPU-n belül az utasítás végrehajtása a következő lépésekből áll:

1. A memóriacímről beolvassa az ott tárolt számot egy belső tárolóba.
2. Hozzáadja az utasításban szereplő számot.
3. Az eredményt kiírja a memóriacímre.

Nyilvánvaló, hogy egy ilyen utasítás hosszú időre megakasztja az utasításcső működését, nagyon lerontja annak hatékonyságát. A sebesség növelése érdekében az ilyen utasításokat el kellett hagyni, innen ered a RISC név (Reduced Instruction Set Computer, azaz csökkentett utasításkészletű számítógép), szemben a CISC (Complex Instruction Set Computer), vagyis összetett utasításkészletű számítógépekkel.

Processzorok jellemző adatai

	VAX 11/780	Motorola 68020	Intel 80386	Motorola 88100	Intel 80860	DEC Alpha 21064
Felépítés	CISC	CISC	CISC	RISC	RISC	RISC
Utasítások száma	304	108	132	51	82	168
Címzési módok száma	16	18	11	7	4	4
Használható adattípusok	14	7	12	7	5	6
Regiszterek száma	16	8	8	32	32	32
Regiszterek mérete	32bit	32bit	32bit	32bit	32bit	64 bit

Mi is a különbség valójában?

Az utasításkészlet összeállításánál a gyors végrehajtás mellett a statisztikát is figyelembe vették, amely szerint a CISC processzorok idejük nagy részében utasításkészletüknek mindössze 10-20%-át használják.

A gyors betöltés érdekében az utasítások hossza — amennyire csak lehetséges — megegyezik a CPU szóhosszával, amely korábban 32 bit volt, ma már egyre inkább 64 bit, azaz két utasítás fér bele.

A bonyolult utasításokat a CISC processzorok egy mikrokódnak nevezett belső program segítségével hajtják végre, amelyet a CPU-ba beépített belső ROM memória tartalmaz. (A mikrokód szerzői joga, illetve használati joga volt a tárgya az Intel és az AMD között évekig tartó pereskedésének.)

A RISC processzorokban nincs mikrokód ROM, minden utasítást közvetlenül, hardveres úton dekódolnak, ami lényegesen gyorsabb. A mikrokód ROM által igényelt tranzisztorokat vagy elhagyták — költségcsökkenés! —, vagy a nagyobb teljesítmény érdekében tárolókat alakítottak ki, kétféle formában.

Egyrészt megnövelték a programok által használható regiszterek számát: míg például a 386/486-os 8 db 32 bites regisztert tartalmaz, a RISC CPU-k minimum 32-t.

A másik lehetőség a processzorban kialakított cache memória, amely kezdetben 2-4 kbájt volt, ma általános a 16/32 kbájtos méret. Gyakran használnak különválasztott utasítás- és adatcache-t, ami azzal az előnnyel jár, hogy egy utasítás adatelérése nem késlelteti az utána következő utasítások betöltését.

A sok regiszter lehetővé teszi, hogy az utasításokkal a memóriához csak adatbetöltéssel, illetve adatkiírással férhetünk hozzá, minden műveletet a belső regiszterekben kell végezni.

A táblázat értelmezése

Táblázatunkban 3-3 CPU-t hasonlítunk össze. A mai típusok kivétel nélkül tartalmazzák a lebegőpontos egységet is (az FPU-t), ezért az utasításkészletben a lebegőpontos számokkal végzett műveletek is benne vannak. A regiszterek közé azonban nem számítottuk be az FPU regisztereit, mivel ezek csak lebegőpontos számok tárolására használhatók.

Az Alpha utasításkészlete lényegesen nagyobb, mint mondjuk a 80386-é, még ha az FPU utasításait tartalmazza is. A RISC ma már nem valami egyszerűsített, a sebesség érdekében lebutított típust jelent, hanem azt, hogy az utasításkészlet kialakításában a gyorsaság az elsődleges.

Bonyodalmak

A jól kialakított utasításcső ugrásszerűen növeli a teljesítményt, alkalmazása azonban némi bonyodalommal is jár. Nézzünk például egy olyan programrészletet, amely feltételes elágazást — ugrást vagy szubrutinhívást — tartalmaz.

```
CLR R2 (Előző utasítás, törli az R2 regisztert)
CMP R1 10 (Feltételvizsgálat az R1 regiszterre)
BZ cím1 (Ugrás a cím1-re, ha R1=10)
ADD R1 02 (Következő utasítás, R1-hez hozzáad 2-t)
```

Az ADD utasítást csak akkor kell végrehajtani, ha a feltétel nem igaz, egyébként a cím1-en lévő utasítás következik. Az ADD utasításcsőbe való betöltésekor azonban a feltételt még nem értékelte ki. Természetesen, ha a feltétel igaz, bekövetkezik az ugrás, a csőből az ADD törlődik, és a cím1-ről kezdi el bepakolni az utasításokat. Ez tetemes idő- és teljesítményveszteséggel jár. Mivel az R2 törlése az ugrás feltételét nem befolyásolja, megváltoztathatjuk a sorrendet:

```
CMP R1 10
BZ cím1
CLR R2
ADD R1 02
```

Ebben az esetben még az ADD betöltése előtt vizsgál. Így, ha a feltétel igaz, a lehívó fokozat idejében tudomást szerez az elágazásról, az ADD-ot már nem tölti be, hanem a cím1-ről veszi az utasítást. Az R2 regiszter törlését mindenképpen végre kell hajtani, az ugrás ténylegesen csak ez után történik meg, ezért hívják késleltetett elágazásnak.

Az utasítássorrend kialakítása — bár nem árt, ha tud róla — szerencsére nem a programozó, hanem a fordító feladata.

RISC-alfajok

Noha a RISC processzorok csak néhány éve terjedtek el tömegesen, az ötletek többsége egyáltalán nem új. A kaliforniai Berkeley Egyetemen 1981-ben a D. A. Patterson és C. H. Sequin által vezetett csoport tervezett először RISC nevet viselő processzort, amelynek kísérleti példányai 1982-ben már működtek. Ez volt a RISC-I, a maga 31 utasításával, 44 500 tranzisztoron, amit hamarosan követett a RISC-II. A technológia hiányosságai miatt nem tudták elérni a tervezett 7,5 MHz-es órajelet, a chip csak 1,5 MHz-en működött — szemben az akkori gyári CPU-k 5-8 MHz-ével. Így az eredmények nem voltak igazán összevethetők, de látszott, hogy az elv jó, érdemes tovább kísérletezni. Patterson néhány ötletet egyenesen Seymour Craytől, a Cray szuperszámítógépeket gyártó Cray Research Inc. alapítójától kapott.

Nem sokkal ezután a Stanford Egyetemen J. L. Hennessy csoportja jelent meg egy RISC típusú processzorral, amelynek a MIPS (Microprocessor without Interlocked Pipline Stages) nevet adta. Ennek továbbfejlesztett változatát gyártotta később a MIPS Computer Systems Inc., elsősorban a Silicon Graphics gépei számára. 1992-ben a két cég Silicon Graphics néven egyesült, a processzorok továbbra is a MIPS nevet viselik.

A viszonylag rövid idő ellenére a RISC típusok is több generációt tudhatnak maguk mögött. A verseny ebben a kategóriában is éles. Ki emlékszik már például az IBM első RISC CPU-jára (a 801-re), és a kereskedelemben kapható, vele épített első RISC gépre, az IBM RT-re? Ma az iparágban mindegyik jelentős cég gyárt RISC processzort vagy ilyen CPU-ra épülő gépet. Itt csak a legnagyobbakat soroljuk fel:

CPU	Cég
Alpha-sorozat	Digital Equipment Corporation
MIPS-sorozat	Silicon Graphics
SPARC-sorozat	Sun Microsystems
Pa-RISC-sorozat	Hewlett-Packard
88xxx sorozat	Motorola
PowerPC sorozat	Apple—IBM—Motorola szövetség

(Ez az assemblerben dolgozókra nem vonatkozik.)

Előnyös oldal

A RISC CPU-k nyilvánvaló előnye a CISC típusokkal szemben a nagyobb teljesítmény. 6-7 évvel ezelőtt sokan nagy reményeket fűztek a CISC processzorok mikroprogramjához, mivel ezzel egyre jobban lehetett közeledni a magasszintű nyelvek utasításaihoz. Volt, aki egyenesen a Pascal nyelv utasításaival vezérelhető processzor megjelenését jósolta.

Egyik kollégánk néhány hónapja egy hetilapban tette fel a kérdést, hogy miért nem fejlesztenek ki olyan CPU-t, amely közvetlenül az operációsrendszer-hívásokat megértő és végrehajtó képességekkel bír.

A technológiai részletektől eltekintve is meglehetősen rázós ez az elképzelés! Külön processzor, sőt gép kellene a DOS-hoz, a Windowshoz, az OS/2-höz, a Unixhoz? Vagy ha például egy olyan új eszközt — hangkártyát, modemet, optikai tárolót stb. — akarok használni, amely a CPU tervezésekor még nem létezett, akkor a CPU-t is cserélni kellene?

Az egyszerűsített utasításkészlet nem jelenti azt, hogy teljesen le kellene mondani a magasszintű nyelvek és operációs rendszerek támogatásáról. A regiszterek nagy száma lehetővé teszi, hogy a szubrutinokhoz, illetve multitaszkos környezetben az egyes taszkokhoz külön regisztercsoportot, ún. regiszterablakot (register window) rendeljünk.

A regiszterek egy része minden program számára látható, a többiből pedig csak az a csoport, amelyet az operációs rendszer a taszk számára engedélyezett. Taszkváltáskor a kevés regiszterrel rendelkező CPU-knál (mint a 486-os) a regiszterek tartalmát (a taszk ún. állapotát) a memóriába kell menteni, illetve a soron következőét betölteni — mindez időigényes. A sokregiszteres RISC processzoroknál csak a programszámlálót kell átváltani, és a taszkhoz tartozó regiszterablakot engedélyezni.

Az egymást hívó folyamatok a CISC CPU-knál a memóriában lévő vermen keresztül adnak át egymásnak változókat; a RISC módszer lehetővé teszi, hogy a regiszterablakok átfedjék egymást. Így lehetnek olyan regiszterek, amelyekhez két folyamat fér hozzá, ez ideális a paraméterátadáshoz. Mindez a CPU-n belül történik, tehát a lehető leggyorsabban.

Hátrányos oldal

A RISC módszerek természetesen nem csak előnyökkel járnak. A hátrányok közül a legkézenfekvőbb, hogy sok funkció, amely a CISC processzorokban egyetlen utasítás, itt csak több utasítással, esetleg szubrutinnal valósítható meg. A tapasztalatok szerint, noha ez konkrét típustól és alkalmazástól függ, ugyanaz a feladat RISC CPU-n átlag 30%-kal hosszabb kódot eredményez. Ez nagyobb memória- és háttértároló-kapacitást igényel.

Hátrányként szokás megemlíteni még, hogy mivel az utasításcső hatékony kihasználása függ az utasítások sorrendjétől, a teljesítményt erősen befolyásolja a fordító. Rossz fordítóval akár 15-20%-ot is elveszíthetünk a lehetséges maximális sebességből.

Csórián Sándor

Amikor a hardverhiba oka szoftverhiba

Miért tévedett a Pentium?

A múlt év végének egyik szenzációja volt a Pentium lebegőpontos számításokat végző egységében (Floating Point Unit) talált hiba. A gyártó Intel szerint ez a felhasználók számára szinte észrevehetetlen, ezzel szemben az IBM olyan súlyosnak ítélte a helyzetet, hogy saját értékesítésében átmenetileg leállította a pentiumos gépek eladását. Kinek volt igaza, és mi is ez a hiba valójában, ezt mutatjuk be röviden — most már utólag, csak a tárgyilagosság és a tanulságok kedvéért. A probléma a lebegőpontos számok osztásához kapcsolódik, és olyan formában jelentkezik, hogy az osztások eredménye bizonyos esetekben lényegesen pontatlanabb a várhatónál.

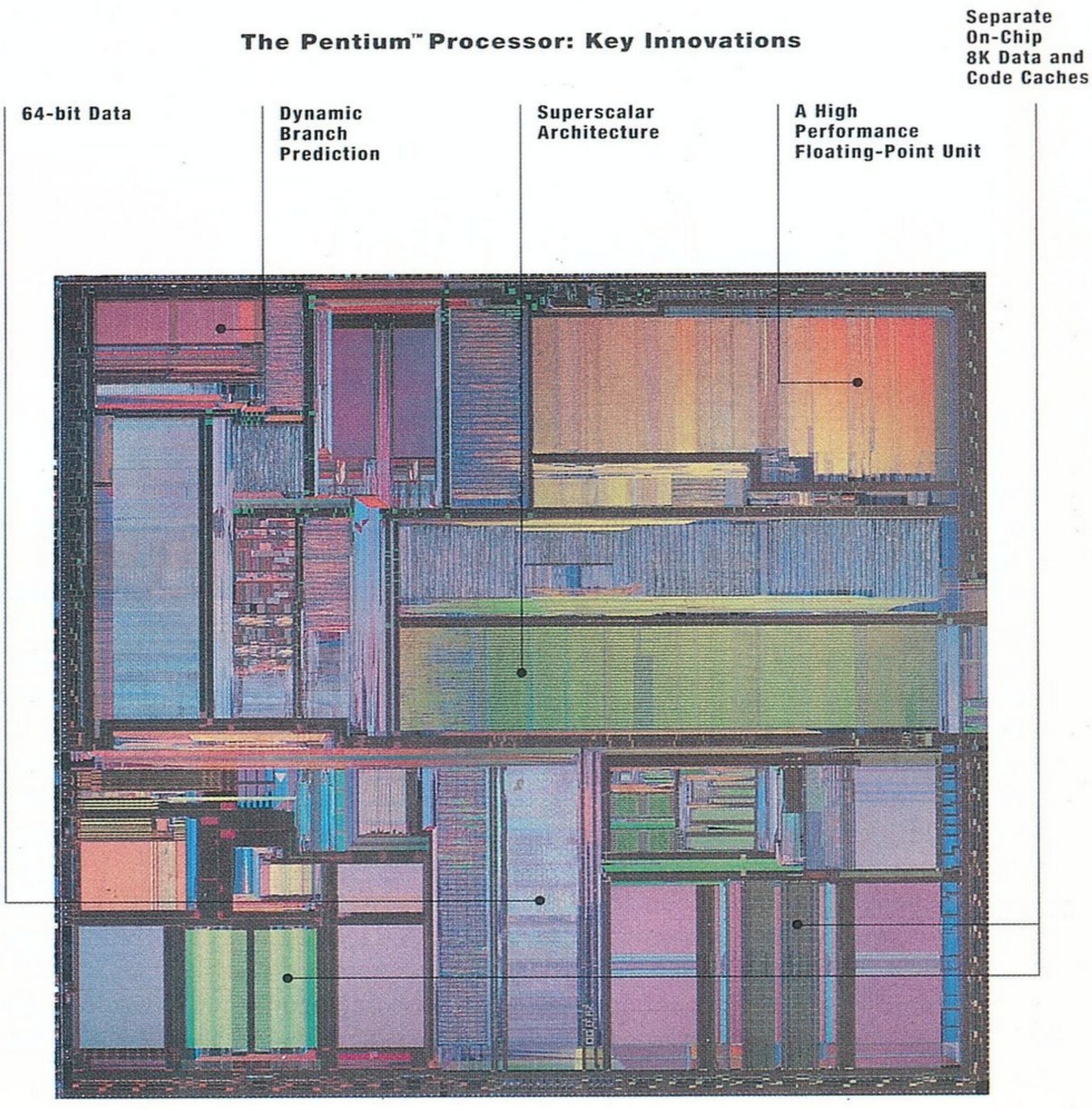
Az IBM ismét kínál pentiumos gépeket, a korábban nagy zajjal leállított értékesítésnél több szakértő emlékeztetett a cég és az Intel-vetélytárs Cyrix között nem sokkal korábban kötött megállapodásra. Eszerint az IBM Cyrix tervezésű 486-osokat gyárt és használ fel, és sorozatgyártás előtt áll a Cyrix Pentium kategóriájú CPU-ja, az M1.

A Pentiumot az Intel már a nagy teljesítményű tudományos, mérnöki és multimédia-alkalmazásokat futtató RISC processzorok versenytársának szánta. Ezért a 486-oshoz képes teljesen új lebegőpontos egységet terveztek, amely azonos órajel esetén 3-5-ször nagyobb sebességű. Lebegőpontos, vagyis nem egész számok osztásakor a 486-osban megvalósított módszer a hányados minden egyes bitjének a kiszámításához egy-egy óraciklust igényel. A Pentium egy új, SRT algoritmusnak nevezett eljárást alkalmaz, amely egy óraciklusban két hányadosbitet szolgáltat.

Az a bizonyos művelet

Ehhez egy gyorsítótábla szükséges, amelyet beépítettek a processzor mikrokód ROM-jába. A hányadost — mint sok mást, például a szinusz- vagy a logaritmusfüggvényt — egy iterációs sorozattal keresi. Az iteráció közbenső lépéseihez kell a táblázat, amelyből az egyre közelítő eredmény bitsorozata alapján választja ki a megfelelő elemeket. A táblázatot legegyszerűbb, mondjuk, egy adószámítást megkönnyítő táblázathoz hasonlítani, amelyben számolás nélkül, egyszerű kereséssel megállapítható az adott jövedelemhez tartozó adó. A kapott eredmény pontossága jórészt a táblázat számainak a pontosságán múlik.

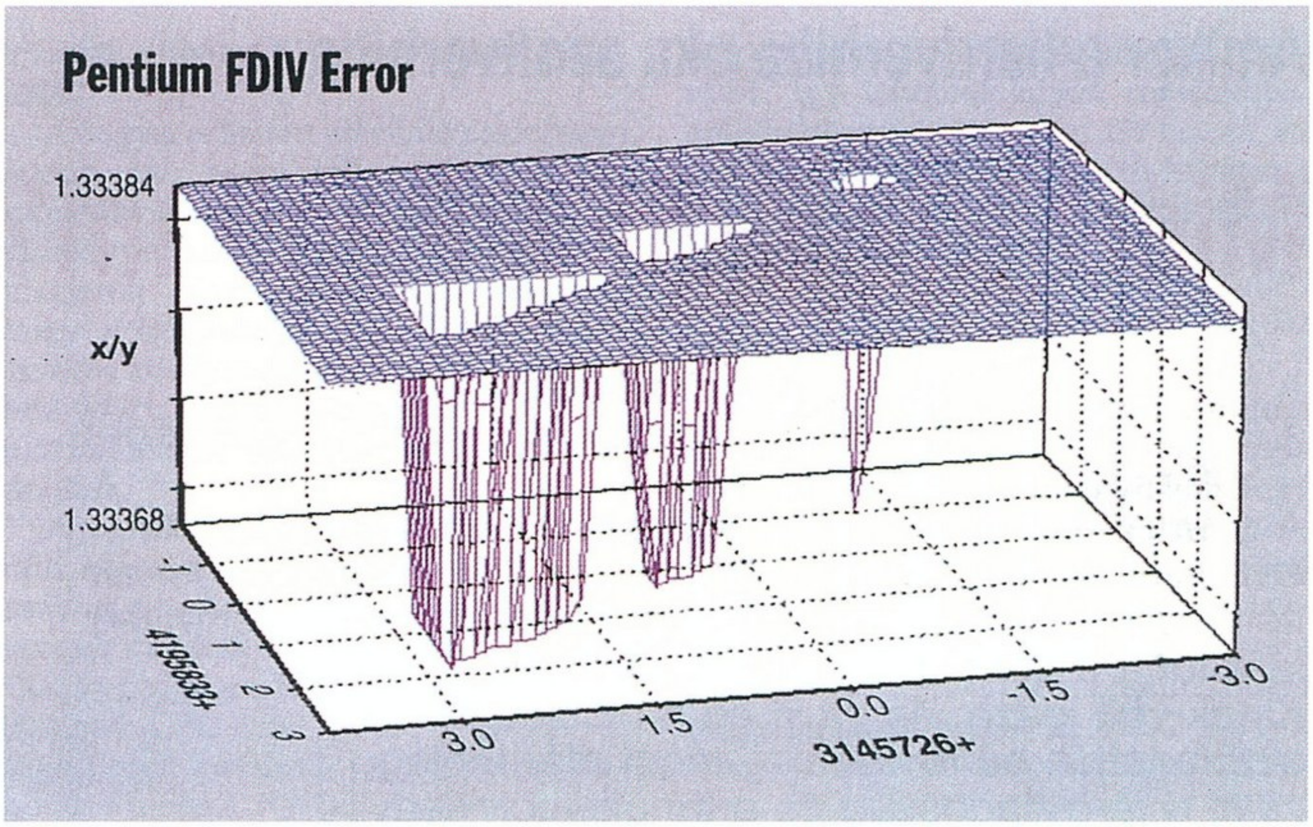
A Pentiumban lévő tábla mátrixa 2048 cellából áll, amelyből jelenleg 1066 tartalmaz értéket. Ezek –2 és +2 közé eső egész számok, beleértve a 0-t is. A tervezéskor egy C nyelven írt rutint használtak az értékeknek egy programozható áramkörbe való letöltéséhez. A rutin hibája miatt azonban az 1066 számból 5 db +2-es értéket kihagyott. A letöltést később senki sem ellenőrizte, a táblázat ezeken a helyeken üres maradt, ami azt jelenti, hogy számításkor innen 0-t kapunk vissza. Ez az eredmény pontatlanságában jelentkezik, aminek előfordulását és mértékét azonban nem lehet előre egyértelműen jelezni. A műveletek többsége elkerüli azt a bizonyos 5 cellát, tehát az eredmény pontossága megfelel az elvárásnak. Amennyiben mégis szerepük lesz ezeknek, az eredmény pontossága tényleg



csökken, és ráadásul — mivel iterációról van szó — a hiba nagysága attól függ, hogy a számítás hányszor érintette az egymást követő lépések során ezeket a cellákat.

Decimális számnál a hiba legroszszabb esetben a mantissza 4. értékes jegyéig nőhet, ami nem azonos a negyedik tizedesjeggyel: a tizedespont helyét a kitevő értéke adja meg, és bárhol lehet a mantisszán. A legroszszabb eset valószínűsége 1:360 milliárdhoz. Általában — már amennyire ezt átlagolni lehet — decimális számnál a 9-10. jegyben jelentkezik, ennek a valószínűsége már nagyobb, 1:9 milliárdhoz.

Mivel az algoritmus a táblázatból a közelítő eredmények alapján választ, a hiba nem kötődik adott osztandóhoz vagy osztóhoz, hanem csak bizonyos számpároknál jelentkezik. Ezek egyértelműen azonosíthatók, ezért is készülhettek bizonyos programokhoz hibakikerülő rutinok. A hiba nyilvánvalóan elsősorban az osztáshoz kötődik, de erős a gyanú, hogy mivel más műveletek is használhatják a gyorsítótáblát, esetleg más utasítások eredményét is befolyásolja. Erről azonban — tudomásom szerint — nincsenek tényszerű adatok, biztosat csak az Intel CPU-tervezői mondhatnának (ha akarnának).



A Byte magazin fenti 3D ábráján grafikusan is modellezték a „lyukas" számrácsot

Mi mennyi?

A legtöbb vitát nem a hiba ténye, hanem annak súlya váltotta ki. Az Intel szerint a hibába a tipikus felhasználó 27 000 évenként egyszer, míg az IBM szerint akár 24 naponként is beleszaladhat. Mindkét értelmezés igaz lehet, csak az a kérdés, milyen is az a tipikus felhasználó.

Az IBM szerint a hibás párok jóval gyakoribbak, mint ahogy azt az Intel állítja. Ebben V. Pratt (a Stanford Egyetem kutatója) elméletére támaszkodik, amely szerint a pontatlanság miatt a CPU sokszor az eredményként kapott — és a normális esetben egésznek adódó — számokat lebegőpontosként kezeli, ami növeli a hiba előfordulásának esélyét.

Véleményem szerint a hiba jelentkezésének gyakorisága, mivel a CPU megjelenése után több mint egy évvel derült csak ki, nem lehet valami nagy. Az új processzorokat nemcsak az Intel vizsgálja, hanem saját érdekében teszteli az összes komoly rendszergyártó is. A 486-osban is volt hiba, amelyet a Compaq szakemberei találtak meg, de még a tesztpéldányokban. Ezt a sorozatgyártás előtt kijavították, kereskedelmi forgalomba így nem került hibás 486-os.

Tény, hogy az Intel jóval korábban is tudott a Pentium hibájáról, és szándékában volt azt — valószínűleg bejelentés nélkül — kijavítani, csak hát azok a fránya gyártási költségek...! A hiba nyilvánosságra kerülése után a cég először felajánlotta, hogy a felhasználóval való konzultáció után kicseréli a processzort egy hibamentes, új sorozatból származóra. Aztán egy 1994. december 20-án kelt bejelentésben elnézést kért valamennyi érintett felhasználótól, és vállalta, hogy minden faggatózás nélkül, kérésre bárkinek kicseréli a korábbi Pentiumot egy hibátlanra. A kedélyek megnyugodhattak...

Csak objektíven?

Decemberben kedvenc tévés hírműsoromban is foglalkoztak a Pentium-hiba körül dúló vitával, és a laikusok számára rögtön demonstrálták is ennek mibenlétét, mégpedig a következő módon:

A képernyőn megjelent a Windows számológépe és a magyarázó hang kíséretében beírtak egy tízmilliós nagyságrendű számot, amelyet elosztottak egy milliós nagyságrendűvel. Majd az eredményt az osztóval visszaszorozva megállapították, hogy a kapott szám nem egyezik meg a kiinduló adattal. Snitt.

A bemutatóhoz mindenképpen kívánkozik egy rövid gyakorlati és egy hosszabb elméleti megjegyzés.

1. Innen elsősorban a Windows „fedélzeti" számológépéről és nem a processzorról szerezhetünk benyomást.

2. Számológépben, számítógépben és bármilyen digitális ketyerében az adott szerkezet képességeitől függetlenül kétféle szám létezik: pontosan ábrázolható és pontosan nem ábrázolható. Az utóbbiak vannak lényegesen többen, gondoljunk csak a végtelen tizedes törtekre, amelyeknél a processzor kényszerűségből kerekít, és a kerekítés elkerülhetetlenül hibát tartalmaz. A kérdés csak az, hogy mekkorát.

Egy olcsó, egyszerű zsebszámolóba beütve az 1/3 * 3 műveletet, az eredmény 0.99...999 és nem 1. Komolyabb zsebszámolók azonban a fenti műveletre 1-et adnak eredményül, de ez csak a kerekítésnek köszönhető, továbbá annak, hogy belül a kiírt számnál több értékes jeggyel dolgozik, egy 12 jegyet kijelző gép aritmetikája mondjuk 15 vagy 16 jeggyel számol. Így a kerekítés az esetek többségében nem okoz észrevehető hibát. Kis próbálgatással azonban kitapasztalhatjuk ennek a korlátait. Egyébként jó képességűnek ismert kalkulátorommal például a 2-ből kiindulva 4-szer tudok egymás után úgy gyököt vonni (azaz 8. gyököt kiszámolni), hogy ezután ismételt négyzetre emelésekkel újra 2-t kapjak eredményként. Az 5-szöri négyzetgyökvonás (azaz a 10. gyök) után elvégzett inverz művelet azonban az eredményt már 2.000000001-nek hozza ki... Egyik kollégám gépén már 9-szer lehet a gyökvonást úgy megismételni, hogy még 2-t kapjunk vissza.

Nagyon jó, ha szakmánk eredményeit vagy gondjait a kívülállók is megismerik, de talán felesleges mondani, hogy a fenti bemutatónak semmi köze nem volt a Pentium hibájához.

Csórián Sándor

„A kezdők imádni fogják..."

A tömörítők keleti csillaga

Az orosz programozók a szakmai elzártság megszűntével valósággal rászabadultak a világra. Mint minden esetben, amikor a szűkös pénzügyi és hardverlehetőségeket munkával, ötletekkel, tudással kell pótolni, most is muszáj-Herkulesek születtek. A PKZIP, az ARJ és az idén Hollandiában feltűnt Ultra Compressor II árnyékában Eugen (azaz Jevgenyij Nyikolajevics) Roshal piacra dobta RAR névre keresztelt, igen jó tömörítőprogramját. Ez amellett, hogy hatékonyan zsugorít, és kellően gyors, más programokból hiányolt szolgáltatásokat is biztosít egyetlen, mindössze 82 kbájtos .EXE fájlba integrálva.

A kezdők imádni fogják a RAR-t egyszerűsége és könnyű kezelhetősége miatt, nem is beszélve a tömörítési hatékonyságról. A törpítőmesterek a tömörítési arányt és a szeletelés területén hozott újdonságait fogják méltányolni. A programozói lelkületű nyúzerek nyomon követhetik a képernyőn a be- és kipakolás mozzanatait, s emellett a fájlok és opciók kiválasztása semmivel sem bonyolultabb teljes képernyős, interaktív üzemmódban, mint a jó öreg Norton Commanderben.

Dokument jeszt

Először is érdemes a dokumentációról szólni. Szeptember elején került a kezembe az augusztusi keltezésű, 1.52 verziószámot viselő változat, ezúttal ékes francia nyelvű dokumentációval egy RAR1_52F.EXE névre keresztelt önkibontó csomagban. A program maga hál'istennek továbbra is angol nyelvű, csak a kísérőszövegeket fordították le. Szerencsémre olyan szöveg is volt benne, amelyet elfelejtettek franciára fordítani. Egetverő újdonságokat nem hozott az 1.50.2 béta-változathoz képest, csak kisebb kiegészítéseket, hibajavításokat. Ezek közül néhány:

— A solid típusú archívokat is lehet módosítani.

— Valamelyest csökkent a program memóriaigénye.

— Két új kapcsoló (–tk és –tl) szolgál az archív fájl dátumának karbantartására.

— Az SFX modul két kapcsolót kapott (–t és –v) az öntesztelésre és a tartalomjegyzék kilistázására.

— Végre az itt is megjelenő –x kapcsolóval ki lehet zárni a műveletekből fájlokat, hasonlóan a többi tömörítőhöz.

— A DOS shell menüből hívható (ALT-J, még az 1.51-ben jött).

— Hangeffektusok lehetősége konfiguráláskor.

— RAR.LOG-ba irányítható hibanaplózás állítható be a konfiguráláskor.

Az 1.53-as verzió Németországból érkezett, feltörve. A Crack lehetővé teszi az AV kód felrakását a RAR archívokra (kipróbáltam, tényleg működik). Lényegi javulást, valódi újdonságokat még nem sikerült a program 1.53-as verziójában felfedeznem. Az egy említésre méltó dolog, hogy az archív komment forrása most már a fix helyű „=<fájl>" paraméter mellett a „–z<fájl>" kapcsolóval is megadható. Ezzel nincs röghöz kötve, a többi kapcsolóhoz hasonlóan bárhol lehet a parancssorban, nem csak a sor végén.

Teljes képernyős üzemmód

A RAR három fő módon indítható. A legegyszerűbb, ha minden paraméterezés nélkül csak a program nevét beírva indítjuk a tömörítőt: „RAR". Ekkor az első ábrához hasonló képernyőt kaphatunk, s teljes képernyős üzemmódban dolgozhatunk. A Norton Commanderhez hasonló keretrendszer önmagáért beszél, a szóköz, az INS és a szürke + gombokkal jelölhetünk ki fájlokat, fájlcsoportokat. A lehetséges műveleteket a funkciógombok segítségével végeztethetjük el, egyes esetekben az ALT gombot is igénybe véve. Ha valaki nem emlékszik, mi mire való a RAR gombjai közül, az nyomja le az F1 gombot. Az F1 gomb lenyomásakor a második ábrán látható, igazán áttekinthető és tömör két képernyőnyi helpet kapjuk. Sajnos a help akkor is két képernyőn marad, amikor nagyfelbontású (50 soros) üzemmódban dogozunk, amit pedig ismer és használni is tud a program. Talán a következő változatban erre is figyelmet fordít majd a szerző.

Senki ne lepődjön meg azon, hogy a RAR fekete-fehérben indul. Ez csak az alapértelmezés, amit az ALT-C kombinációval válthatunk át színesbe. Egyéb beállítások is elvégezhetők az ALT billentyű segítségével, s a beállítások végén az ALT-S kombinációval az aktuális beállítások rögzíthetők egy RAR.CFG nevű fájlban, amelyet a program a RAR.EXE mellé helyez. Éppen ezért ne indítsuk írásvédett lemezről a programot, ha konfigurálni is akarjuk.

Ha a számítógépet a RAR indítása előtt nagyfelbontású üzemmódban használtuk, a RAR ezt érzékeli, és szintén 50 soros módban jelentkezik. Hogy a zsúfoltabb könyvtárakban való közlekedést, keresést meggyorsítsák, a RAR figyeli a billentyűzetet. Így az S gomb lenyomásakor az első S-sel kez-

A hiányzó DLL-ek

Márciusi és áprilisi lemezmellékletünk megjelenését követően „izzottak" a vonalaink. Az ok mindkét esetben a mellékleten közreadott windowsos programok DLL-hiánya volt. A Visual Basicben írt programokhoz fejlesztőik legtöbbször „elfelejtik" mellékelni a VBRUN*.DLL fájlt (mondván, az már úgyis megvan, valamelyik program biztosan generálta), enélkül viszont a program nem hajlandó elndulni. Segítendő azokon, akik szeretnének utólag is megismerkedni az Exitwin, illetve a Winmoney programokkal, mostani lemezmellékletünkön becsomagolva elhelyeztük a VBRUN300.DLL, a SETUPKIT.DLL és a VER.DLL fájlokat. Ezek a DLL#.EXE önkicsomagoló tömörítvényből bonthatók ki, és másolhatók át a WINDOWS\SYSTEM könyvtárba. (A VER.DLL része ugyan a Windowsnak, de nem minden installálási mód esetén kerül bele a rendszerbe). A kényelmetlenségekért elnézést kérünk a programszerzők — és Billy Gates — nevében is.

dődő fájl- vagy könyvtárnévre ugrik. Ha szükséges, betűnként sorban beírhatjuk a keresendő fájl nevét, amíg el nem érjük. A beírt karakterek a képernyő alján a státussorban sárga színnel jelennek meg. A könyvtárak közötti mozgás is egyszerű: rá kell állni a választott könyvtárnévre, majd jöhet az Enter. A .. (két pont) itt is az előző könyvtárba (szülőkönyvtárba) visz.

Archívok kezelése

A RAR második üzemmódja, hogy közvetlenül betekintünk egy RAR, ARJ, ZIP vagy LZH archívba. Kétféle módon tehetjük ezt meg: vagy a „RAR EN archív_neve", vagy a „RAR archív_neve" paranccsal. Igény szerint akár a Norton Commander NC.EXT állományában elhelyezett „RAR EN !.!", illetve „RAR !.!" parancsokkal közvetlenül is beleléphetünk. Ebben az üzemmódban sok esetben kiválóan helyettesítheti az ARCVIEW, ZIPVIEW programokat vagy egyéb keretprogramokat. A RAR archívba belépő EN parancsa a fentiek szerint minden további nélkül elhagyható, én azonban azt ajánlom, használjuk, hátha a következő verzióban megint kötelező lesz. (Volt már ilyenre példa.)

A tömörített fájlokba a RAR saját kukkantójával belenézve ASCII és HEXA formában jeleníthetjük meg a kiválasztott fájlokat, de mód van az F9 (Option) gomb megnyomása után egy külső kukkantó (external viewer) megadására is. Itt csak a program nevét kell megadni a szükséges kiegészítőkkel, a RAR a megjelenítendő fájl nevét automatikusan továbbadja. Ezután sem vagyunk egyetlen megjelenítő használatára korlátozva, hiszen az F3-ra a külső, az ALT-F3-ra a saját kukkantóval nézegethetjük az aktuális könyvtárban vagy a tömörített fájlokban kiválasztott állományokat. Szükség esetén, ha egy átnézendő tömörítvényben több különböző típusú fájl található, akár menet közben is lehet új külső nézőprogramot definiálni.

A RAR csak akkor tud a listázáson kívül mást is kezdeni a nem saját, RAR típusú tömörítvényekkel, ha az ARJ, az LHA és a PKZIP/PKUNZIP, valamint a ZIP2EXE programok elérési útvonalon (path-on) vannak. Használatára jó példa, ahogy a ViruScan 211E csomagjába nézhetünk bele.

Sajnos a RAR program gyengéje, hogy a jelenlegi verzió sem képes az önkibontó archívokba belenézni, csak ha az a RAR saját önkibontó tömörítvénye.

Az archívkészítők új hulláma

A hagyományos tömörítők (ARJ, PKZIP, LHA stb.) egyszerre csupán egyetlen fájl tömörítésével foglalkoznak, s nem veszik figyelembe, ha egy bájtsorozat egymás után több fájlban is megismétlődik. Az Ultra Compressor II és a RAR ezzel szemben képes a becsomagolandó állományokat egyetlen fájlként kezelni (ezt emlegetik a nagygépesek TAR and COMPRESS elvként). A tömörítő ilyenkor csak hivatkozik az ismétlődő részre, a különbséget, az eltérést pedig tömöríti. E megoldásnak természetesen méretbeli korlátai is vannak. Elsősorban olyankor vezet igen jó eredményre, ha egymástól alig különböző fájlokat kell csomagolni, mint például a kiküldött körlevelek archiválási céllal történő tömörítése esetén.

A másik jelentős lépés, hogy kibontáskor az eredeti útvonalon vagy az aktuális könyvtáron kívül is kibonthatók a kijelölt fájlok, még ha bonyolult könyvtárszerkezetet kell is létrehozni. Külső programok használatakor ez engedélykéréssel jár, saját archív esetén az új könyvtár létrehozása automatikus. Ilyen szolgáltatást a holland Ultra Compressor II és a RAR mellett csak egyes speciális archívkezelő keretprogramok nyújtanak.

Az ARJ hozta be 1991-ben a több lemezre kiterjedő, úgynevezett szeletelt archívok készítésének lehetőségét. Azóta más fejlesztők is ellátták programjaikat hasonló szolgáltatásokkal (PKZIP, HPACK stb.), ám egy funkcióval adósak maradtak: az önkicsomagoló fejléccel ellátott szeletelt archívok létrehozásával. A RAR ezt is megvalósította. A szeletelt tömörített archív első darabja egy EXE kiterjesztésű fájl. Ezt mint közönséges programot elindítva az első szeletre kerülő, nem egészen 7,5 kbájtos EXE programfej gondoskodik a soron következő szeletek bekéréséről és kibontásáról. Az 1.52-es változattól ez az önkibontó fej a -t kapcsolóval öntesztet végezhet, a -v kapcsolóval ki tudja írni az SFX saját tartalomjegyzékét, így a felhasználó még a kibontás előtt eldöntheti, hogy kell-e neki, ami benne van.

A szeletelő SFX technikával a kereskedelmi szoftvercsomagok terjesztői megszabadulhatnak attól, hogy külön kicsomagoló programot kelljen mellékelniük. (Ehhez azonban nekik is meg kell vásárolniuk a regisztrált RAR-változatot!)

A RAR az AV szignó mellett archív kommenteket és fájlkommenteket is elhelyezhet az elkészített (akár önkibontó) tömörítvényeken. Ez önmagában nem lenne különlegesség, de az már igen, hogy a kommentfájlokban (amelyek méretét illik 16 kbájt alatt tartani) színező és kurzorpozicionáló ANSI parancsokat, szekvenciákat is el lehet helyezni. A RAR ugyanis támogatja ezek használatát, még akkor is, ha az ANSI.SYS-t NEM telepítettük a gépre!

A billentyűzet átdefiniálását végző parancsokat a RAR nem támogatja, tehát védettséget biztosíthat az ANSI-bombák ellen, s nem kell attól félni, hogy egy jóakarónk a kommentekbe rejtett ANSI parancsokkal goromba rejtett műveleteket definiál számunkra.

Nagy Gábor

A RAR további fogyatékossága, hogy a megjelenítésnél és a kibontásnál a kijelölt fájlok után keresve az egész archívot kibontja a memóriába, amíg a kiválasztott fájlok is sorra nem kerülnek. Ez meglehetősen lelassíthatja a munkát, különösen akkor, ha sok fájlt tartalmaz az archív.

A kísérő dokumentáció szerint a RAR regisztrált változata egy példányra 35 USD. A regisztrált felhasználó saját, maximum 60 karakteres szignója minden -AV kapcsolóval készített RAR archívon rajta lesz, s egy éven keresztül kap a szerzőtől támogatást és frissítéseket. A BBS-eken a RAR-nak feltört változatai is keringenek. Az eredeti RAR.EXE LZEXE-vel van tömörítve, a rosszul átvakart, magát regisztrált változatnak tituláló RAR pedig PKLITE-tal. A végeredmény: hiába a „Registered version" felirat, az nem tesz fel AV szignót, és más, csak a teljes változatban meglévő szolgáltatások sem működnek. Az 1.52-es verziótól kezdve van az eredeti csomagban egy UNRAR nevű kicsomagoló program is, amely freeware, azaz szabadon felhasználható, és minden fizetési kötelezettség nélkül a tömörített .RAR fájlok mellé tehető. (A leírás folytatása a lemezmellékleten, ahol shareware változatban maguk a programok is megtalálhatók.)

Nagy Gábor

A Georgikon szelleméhez méltó törekvések

A „legeslegfontosabb” multimédia

A „legeslegelső” lézerképlemezes interaktív videorendszerek megjelenése óta nyilvánvaló, hogy a multimédia-termékek és -technológiák „legeslegtermészetesebb” alkalmazási területei között a „legeslegfontosabb” az oktatás, a tanulás. Mégsem ez „futott be” — legalábbis mennyiségi szempontból nem — az azóta eltelt több mint tíz év alatt. Inkább a szórakoztatóelektronikai irányzatot kiszolgáló tömeges produktumok fejlesztését, készítését és forgalmazását táplálták bő források. Mostanában talán mégis változik a helyzet.

Az egyetlen dolog, ami vigasztaló a multimédia-piacnak a könnyebben járható utak választásával összefüggő torzulásában — vagyis abban, hogy jóformán csakis „karneváli maszkban” lépett fel eddig —, a következő. Nem ok a fellélegzésre, de a startvonalat tekintve mégiscsak relatíve kedvező előzmény, hogy ez ellen sehol a világon nem sikerült túl sokat tenni.

„Menedzserek” és „befektetők”

Nem vagyunk tehát itthon sem lemaradva: az oktatásban jogosan elfoglalandó szerepe a multimédiának még az európai egyetemeken sincs megkoreografálva, általában mindenütt csak a „belépő” jelenetnél tartanak. Ott viszont mi is tartunk már.

Mindez egy Keszthelyen, a bicentenáriumához közeledő Georgikon utódjának szervezésében megtartott workshop résztvevőinek másfél napi eszmecseréje kapcsán derült ki világosan. Annak köszönhetően, hogy jóformán mindenki ott volt, aki ma a felsőoktatás körül és berkein belül ebben a „többfonalú” eszközben utazik. A mindenki úgy értendő, hogy a legfontosabb fórumoktól általában volt jelen képviselő, vagy ha valamely „formáció” személyesen nem is tudta delegálni „előmunkását” e munkaértekezleti alkalom teremtette együttmunkálkodásra, azért hírt adhatott nézeteiről, eredményeiről. Így például Racskó Péter, az Állatorvostudományi Egyetem tanszékvezetője, aki egy FEFA-projekt során felmerülő kérdésekről és tapasztalatokról szólt, előadásában nyilvánvalóan a Budapesti Műszaki Egyetem, a Budapesti Közgazdaságtudományi Egyetem, az ELTE Computer Algebra Tanszék és az MTA SZTAKI erőfeszítéseit is közvetítette a közös munkáról szóló beszámolón keresztül.

Érdemes lenne a hazai oktatási multimédiatermék-fejlesztések és koncepciók viszonylag jó pozícióját a világszinthez képest megtartani, mert ez az üzlet még csak most fog igazán beindulni. Máskor is élen jártak már hazai szakembereink valamely jelentős műszaki vagy tudományos folyamat kibontakoztatásában — főleg az elején és még egy darabig. Azután valahogy mindig leszakadtunk — hol így, hol úgy, hol ezért, hol amazért. És mindannak ellenére, hogy ezt a szakmai társadalom nagyon nem akarta, és a legkitűnőbb szószólók jó előre jelezték a következményeket azoknak a döntéshozóknak, akiken a termék-, műhely-, ágazatjövő (és velük együtt sokszor igen keményen az embersorsok) alakulása végül is múlott.

Az oktatás minőségének, eszközeinek, hatékonyságának lépéstartása viszont egész generációk sorsára hat — tudjuk; ezért még súlyosabb az a követelmény, amelyet a munkaértekezlet végén a résztvevők önmaguk számára is megfogalmaztak: nem szabad feladni semmit abból a pozícióból, amelyig felsőoktatásunk multimédia-projektjei eljutottak.

Didaktikai alapozás

Noha azon még „belterjes” körben is kialakulhattak volna komoly viták, hogy ki, mit nevez pontosan multimédiának, és ezen a vitán belül kik kép-

viselik az álmokat, és kik a „földhözragadtak", a legfontosabb összetevők és célok egyetértést kovácsoltak a szakemberek között.

— Az emberi érzékelésben, a környezet megismerésében kiugróan a vizualitás jelentősége a legnagyobb. Mivel a multimédia konkrétan a számítógépes oktatásban (CAE) találja meg legkönnyebben a fészkét, ezen keresztül a vizualitás ereje fokozza az oktatás hatékonyságát. Mármint ha van a multimédiát mint technikai eszközt profi módon, megfelelő minőségben hasznosító oktatási anyag. (Angolnyelv-oktatót például már *igazit is láthattunk.)*

— A kidolgozott eszközök az oktatás és tanulás különböző fázisaiban másmás célra legyenek megfelelők, fő funkciójuk szerint sokfélék legyenek: az önálló tanulást is tegyék lehetővé; ismeretmegerősítő szerepük dominál-jon; demonstrációként egészítsenek ki más módszereket; csak illusztrációként hasznosuljanak. Természetesen igény lehet arra is, hogy ezek a szempontok párhuzamosan érvényesülhessenek a különböző szituációkban. (Láttunk olyan anyagot, amelyben például a vizsgáztatás támogatása volt a legerősebb koncepcionális és szerkesztési elv, a legfőbb cél.)

— A multimédia képességei más, hagyományos eszközökkel együtt is kiaknázhatók, így az oktatóbázisok bizonyos specialitásaik függvényében ugyan, de mindenképpen át fognak alakulni. Előre látható, hogy önálló tanulólaborok épülnek majd ki, s ezt hamarosan a gazdasági környezeti hatások is serkentik. (A hálózatok technikailag a decentralizáltan jelentkező oktatási-továbbképzési igények és a központi fejlesztési, üzemeltetési „felügyelet" mellett máris adják az elvi lehetőséget a megoldásra. A távoktatás például a „multimédiára van kitalálva".)

— A multimédiára épített oktatórendszerek fejlesztésének alapos elemzésre és igazi csoportmunkára kell támaszkodnia, amelyben didaktikai szakemberekre is szükség van a technikai, művészeti, alkalmazási, szakterületi stb. specialisták mellett.

— A fejlesztést előkészítő vizsgálatok abból indulnak ki, hogy az ismeretanyag mennyiben igényli a multimédiás kibontást. (Csak prezentációt-e, vagy szimulációt is; a gyakoroltatást miként és milyen „ritmusban" stb.) Sőt: egyáltalán megéri-e a befektetést a hatékonyság javulásában várható eredmény. Fontos a fogadókészség felmérése is, annak jogi és informatikai vonatkozásaival együtt. Ugyanígy mérlegelendő, hogy milyen az anyag frissítésének szükségessége és lehetősége.

Dedikáció

Talán az alapításának bicentenáriumát már kétévnyire megközelítő Georgikonnak — mint a kitűnően szervezett, alkotó hangulatban zajló workshop rendező gazdája elődjének — a hagyományai is erősítették a meggyőződést a jelenlévőkben: érdemes intenzíven foglalkozni a *Multimédia alkalmazása az oktatásban* témával. Nagyon jó lenne ugyanakkor másoknak is átvenni az ilyen munkákban már jó ideje jártasságot szerzett kutatók, fejlesztők, kísérletezők szellemi attitűdjét: a felelős útkeresést, amely az oktatási és képzési területen dolgozó értelmiség legjavát mindig is jellemezte.

Jakab Ágnes

Karaktersorozatok cseréje programmal — II.

Megérteni, megírni, működtetni

A cikk első felében a feladatok értelmezéséig és részletezéséig jutottunk. Most a megvalósítás kerül előtérbe. A szekvenciafelismerő módszerek két legáltalánosabb megoldása, hogy a program vagy az állapotterében tárolja a bemeneten érzékelt sorozatot, vagy pedig egy léptetőregiszterben lépteti az adatokat, és ott végzi az összehasonlítást.

A mi esetünkben a léptetőregiszteres módszer a megfelelő, mert a léptetőregiszter jól illeszkedik a konverziós szabályokhoz (a bal oldalt kell összehasonlítani a léptetőregiszter tartalmával).

A regiszter egyszerűen az input verem tetején helyezhető el, a mi esetünkben azonban fizikailag nem pontosan a verem tetején, hanem mint valóságos léptetőregiszter van megvalósítva az átalakító eszközön belül; logikailag azonban a léptetőregiszter az input verem része.

Az output szöveget a program hozza létre, tehát a kezelése is tőle függ. Szeretnénk azonban megtartani az input fájl azon kellemes tulajdonságát, hogy szekvenciálisan olvasunk belőle, és nem írhatunk bele, hiszen egyrészt nem szabad megváltoztatni, másrészt egy szekvenciális állományba való beszúrás vagy törlés jelentős adatmozgatással jár, ami nagyméretű állomány esetén kellemetlen, és óriási időveszteség. Ezért az input oldalon a feladat megoldása mindenképpen a verem emulációja (az input verem az aktuális pozíciót követő részt jelenti).

A pontos „teendők"

Az átalakító eszköznek tehát nemcsak a fájlokat kell veremként kezelnie, hanem el kell végeznie a léptetőregiszterrel kapcsolatos karbantartásokat is. Először is: a léptetőregisztert kezdetben fel kell tölteni, lehetséges azonban, hogy az input szöveg eleve rövidebb, mint a léptetőregiszter hossza. Ha a léptetőregiszter fel van töltve, nem okoz különösebb nehézséget a karbantartása, még abban az esetben sem, amikor nem egyedi karaktereket, hanem karaktersorozatokat kell belőle kivenni.

Az igazi nehézség akkor adódik, amikor az input fájl végéhez közeledve a hosszabb feltételek már nem illeszthetőek, a rövidebbek viszont még igen. A megvalósítás úgy történt, hogy az összehasonlítás során az eszköz a szabályok feltétel oldalát mindig a léptetőregiszter az ábra szerinti bal oldalára illeszti. Ezért a léptetőregiszter hossza nem fix, csak maximalizálva van a leghosszabb feltétel hosszára, egyébként pedig dinamikusan van nyilvántartva.

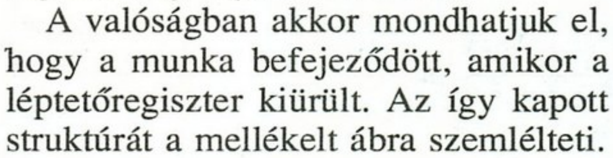
Az előbbieket összefoglalva tehát azt mondhatjuk, hogy az input oldalon található verem valójában három részből áll össze (alulról felfelé nézve):

— Az Input1 sorozat az állományvégtől az aktuális pozícióig (tehát visszafelé).

— Az Input2 emulált verem (ahova visszaírunk) az aljától a tetejéig.

— A léptetőregiszter a bal szélső regiszteréig teljes hosszában.

A valóságban akkor mondhatjuk el, hogy a munka befejeződött, amikor a léptetőregiszter kiürült. Az így kapott struktúrát a mellékelt ábra szemlélteti.



Mint azt már említettük, a helyettesítési szabály alkotja az utasítás egyik felét, a másikat pedig a pozicionálás vezérlése. Mivel általában több utasítás (konverziós szabály a hozzá tartozó pozicionálási paranccsal) van egy programban, ezeknek az utasításoknak a precedenciája a felsorolásuk sorrendje lesz.

A teljes utasítás első fele a helyettesítési szabály, amelyet a feltételt leíró sorozat, a „->" operátor és a cserélendő sorozat alkot. A feltételt tartalmazó sorozatnak az elemeit hasonlítja össze az automata a léptetőregiszterben található sorozattal, és a feltételt kielégítő sorozatot — nem pedig a teljes léptetőregiszterben található sorozatot (!) — cseréli ki a cserélendő sorozatra.

Bal és jobb

Eddig a bal oldalt feltételnek neveztük, mert a szabály alkalmazhatósága a bal oldalban leírt feltétel kiértékelésével dől el. Csak konstans szimbólumok esetén egy szabály akkor érvényesül, ha a bal oldal ugyanaz, mint a léptetőregiszter tartalma, vagyis egyenlőségvizsgálatot kell végezni regiszterről regiszterre. Az interpreter azonban több magas szintű szolgáltatást biztosít a kényelmes és hatékony felhasználáshoz. Először is minden regiszterhez megadható, hogy a szabály bal oldalának megfelelő elemével egyenlőséget vagy nem egyenlőséget vizsgáljon. Másodszor lehetséges terminális szimbólumok halmazába, az ún. csoportba való tartozást vagy nem tartozást vizsgálni. Harmadszor pedig lehetséges két regisztertartalom összehasonlítását is megadni, ez például dupla betűk detektálásakor hasznos.

A szabály jobb (output) oldalán azt adjuk meg, hogy mit kell becserélni az adott pozícióba. Itt nyilvánvalóan nem állhat csoport. Amikor megváltozik egy inputbeli szimbólum, akkor ennek helyére kerül be az output megfelelő pozíciójában lévő szimbólum, ami persze lehet az üres szimbólum is. Amikor nem akarjuk megváltoztatni az inputbeli szimbólumot, vagy az input oldalon egy csoport állt (és így nem tudjuk, hogy a csoport melyik tagja volt az), akkor indirekciót alkalmazunk. Ez azt jelenti, hogy az adott regiszter sorszámát adjuk meg, és a regiszter tartalma íródik ki. Indirekciót használva felcserélhető a szimbólumok sorrendje is. Az

output oldalon lehetőség van függvények használatára, a függvénynek az argumentuma nem az adott pozícióban levő inputbeli szimbólum, hanem az egyik regiszter sorszáma, tehát ez is egy indirekció.

A csere után történik meg a pozicionálás végrehajtása. A pozicionálásnál két lehetőség van: vagy visszalépni az inputba, vagy a becserélt sorozat valamennyi részét kiíratni (de a becserélt sorozatnál többet nem).

A program felépítése

Eddig eljutottunk oda, hogy milyen módon képzeljük el a karaktersorozatot cserélő interpreter működését. Tekintsük meg most magát a megvalósítást!

Az interpreter működése két szakaszra osztható, a program értelmezésére, és a programban leírt tevékenység végrehajtására. Ennek megfelelően a főprogramnak két processz van alárendelve, amelyek végrehajtása szekvenciális. Elsőként mindig a programot értelmező processz fut le, ez állítja elő a szekvenciacserélő processzt inicializáló adatállományt, innen a programinicializáló fájl, vagy röviden ini-fájl elnevezése. Az inicializálás után a szekvenciacserélő processz több inputra is lefuttatható; batch üzemmódban lehet ezzel a lehetőséggel élni. A főprogram végzi a két processz paraméterezését és hívását, e két művelet vezérelhető batch üzemmódban.

Ami kell hozzá

A szekvenciacserélő processz futásához szükséges adatállomány a következő elemeket tartalmazza:

— A csoportok tagsági függvényei. Mivel a tagsági függvény minden elemhez 0 vagy 1 értéket rendel hozzá, ezért ez egy bittérkép. Mivel 256 darab bájt van, ezért egy csoport ábrázolásához 256 bit, azaz 32 bájt kell.

— Az egyváltozós függvények értékei listaszerűen felsorolva. A függvények bájtról bájtra képeznek le, és kiértékelésük fordítási időben megtörténik, tehát a 256 darab bájthoz 256 darab függvényérték tartozik, vagyis egy függvény ábrázolása ennyi memóriát igényel. Ennek a pazarló memóriafogyasztásnak az oka a futási idő gyorsítása, hiszen egyetlen indexelt memóriahozzáféréssel megkaphatjuk az argumentumhoz tartozó függvényértéket.

— Az utasítások. Említettük, hogy egy utasításhoz a helyettesítési szabály két sorozata, és a pozicionálást vezérlő parancs tartozik.

— Az input program kerete (frame). Ez két sorozatot jelent, amelyek keretbe foglalják az input szöveget, lehetőséget adva a léptetőregiszter kezdeti feltöltésére, a fájl határainak kezelésére. A két sorozat megadása opcionális.

Az utasítások kivételével a többi három adathalmaz opcionális. A csoportok és függvények definiálásához lehetőség van makrók és tartományok használatára, ezek az elemek azonban nyilván nem szerepelnek a szekvenciacserélő processz futásához szükséges adatállományban.

A konkrét megvalósítás, a hozzá tartozó felhasználói dokumentáció és néhány, általunk használt inicializáló fájl dokumentációja a lemezmellékleten megtalálható, részletes információk ezekből kaphatóak. A program használatához a szerző sok sikert kíván!

Lois László

A tervezés szakértőrendszerei — II.

Dedukció, indukció és...

Ma még eldöntetlen, hogy miképpen lehet legjobban megragadni a tervezést végző ember gondolkodását. A tervezői tudás szemléltetésében is csak az ismert módszerek kombinálása vagy hibrid formák alkalmazása vezethet eredményre. Éppen ez az egyik forrása a fejlesztésben tapasztalható komplexitásnak.

Elvileg deduktív, induktív vagy abduktív megközelítések alkalmazhatók. A deduktív gondolkodáshoz kapcsolódó szillogizmus a tény—szabály—következmény struktúrában könnyen formalizálható. Azok a legegyszerűbb szakértőrendszerek, amelyek ezt a gondolkodási formát modellezik, állításokkal írják le a megoldandó problémát, ezekre alkalmazzák a tervezői (mély vagy felszíni) tudást megtestesítő szabályokat, és előre- vagy hátraláncolással a megoldás létrehozása érdekében szükséges ismereteket. A szabályok — bármennyire természetes és hatékony tudásszemléltetési formát képviselnek is —, önmagukban nem tudják lefedni a tervezéshez kapcsolódó ismeretek teljes körét.

A deduktív feldolgozás eleve feltételezi a lehetséges eredmények ismeretét, ami beszűkíti az ezt alkalmazó szakértőrendszertől elvárható innovativitást vagy kreativitást. A deduktív típusú tervező szakértőrendszereknek szembe kell nézniük a zárt ismeretvilágukból adódó összes korlátozással. A problémát egyetlen, a szakértőrendszerek szempontjából kezelhetetlen döntési szituáció megemlítésével szeretnénk alátámasztani: *azért, mert valakinek az ujjlenyomata egy pisztolyon megtalálható, még nem biztos, hogy ő lőtt a pisztollyal.*

Tervezési feladatok megoldásában alkalmazható szakértőrendszer-vázak

Munkaállomásra

Exsys, Hearsay, Emycin, Knowledge Caft, Loops, Timm, Reveal, ART, SRL+, KEE, S.1, KES, Personal Consultant.

Személyi számítógépre

Insight 2+, 1st Class, Kappa, M.1, Expert Choice, KES II, Expert Ease, Advisor, Pangea, Guru, Personal Consultant, IKE, Wizdom.

A tudás származtatása és szemléltetése

A dedukció lehetséges alternatívája az indukció, amely lényegében magának a tudásnak a származtatására irányul. A tudományban az indukció a jelenségek megfigyelése alapján származtatott elmélet létrehozásának folyamata. Az induktív gondolkodási paradigmát megvalósítani igyekvő szakértőrendszereknek tehát valamilyen formában ismeretszintézist kell végrehajtaniuk. Ezzel kapcsolatban napjainkra már többé-kevésbé körvonalazódtak a lehetőségek, és egyértelművé vált, hogy itt is a szimbolikus tudásszemléltetés korlátaiba ütközünk. A szimbólumok mögötti szemantikai tartalom tényleges megértése nélkül viszont nagyon nehéz értelmes ismeretszintézist megvalósítani.

Az induktív problémamegoldás sajátos problémája az implicit tudás és a plauzibilis tudás bevonása, fejlesztési oldalról pedig a tanulási képesség rendszerbe építése. Az induktív következtetési paradigma hasznosítására többféle végrehajtási módszertant dolgoztak ki. E helyen csak (megtartva a MI-kutatásban kapott megnevezések helyénvaló magyar megfelelőit) a példák alapján való tervezést, a megszorításkezelésen alapuló tervezést, és az üzenetátadó funkcionális ügynökökre alapozott tervezésmódszertant említjük meg.

Az említettek mindegyike valamilyen tervezési prototípus meglétét tételezi fel, ami összevág az indukciónak a jelenségek megfigyelésére alapozott megközelítésével. Az indukció paradigmájára támaszkodó szakértőrendszerek valamilyen extrapolációval jutnak el a megoldáshoz, amelynek során felhasználják a prototípusokra megismert tervezési ismereteket és törvényszerűségeket. Kizárólagosan a dedukcióval kapcsolatban korábban adott kezelhetetlen példánk továbbvitele érdekében, az indukció esetében a problémát a következőképpen lehet szemléltetni: *azért, mert valaki lőtt egy pisztollyal, még nem biztos, hogy az ujjlenyomata azon megtalálható.*

Tanulás, taníthatóság, önfejlesztés

A tervezésben eredményesen alkalmazható szakértőrendszernek nem egyszer létrehozott vagy időszakonként továbbfejlesztett eszköznek, hanem a tudása tekintetében az alkalmazások során gyarapodó rendszernek kell lennie. Ez ellentétben áll a szakértőrendszerekre hagyományosan elfogadott „zárt világ" feltételezésnek. A szakértőrendszereknek a saját problémamegoldási és alkalmazási tapasztalatot is meg kell ragadniuk, vagyis ugyanazt a feladatot később kisebb ráfordítással kell megoldaniuk. Ez a tanulás képességét és lehetőségét tételezi fel.

Az indukcióhoz kapcsolódó tanulást ma még sokkal inkább a taníthatóság lehetősége, mint az öntanulás képessége jelenti. A spontán tanulás megfelelő gépi formáját még nem sikerült reprodukálni. A tanulás tekintetében a példákhoz kapcsolódó és az adaptív tanulási forma örvend kitüntetett figyelemnek — ezek a tervezés szempontjából is jelentősek.

Tapasztalva a deduktív és az induktív következtetési formák kötöttségeit, több kutató és fejlesztő az abduktív tudásfeldolgozást helyezte a középpontba. Ezzel részletesebben egyik következő cikkünkben foglalkozunk.

A tervező szakértőrendszerek nem közelíthetik a tervezést általános értelemben. Akkor lehetnek igazán hatékonyak, ha valamely alkalmazási területen adott tervezési feladattípusok megoldására fejlesztik őket. A tervezésben megkülönböztetünk kereső, átalakító, kon-

figuráló, kombináló és generáló feladattípusokat. Ezen feladatokat nemcsak az irányultságuk, hanem a megoldásukhoz szükséges algoritmikus és/vagy heurisztikus módszerek is elválasztják egymástól.

Tipikus és speciális feladatok

A kereső, átalakító és konfiguráló feladatokban diszkrét keresési tér elemeinek leválogatása, elemzése és összerendezése, a kombináló és generáló feladatokban a termékre és az előállítási folyamatra vonatkozó információk előállítása és szintézise jelenti a rendszertől elvárt funkcionalitás alapját. Az utóbbi két feladat esetében nehézséget az jelent, hogy gyakorlatilag folytonos eloszlású keresési téren kell műveleteket végezni, megszorításokkal behatárolt kompozíciós feladatként.

A hagyományos szakértőrendszerek monoton következtetést valósítottak meg, feltételezve, hogy az új tények nem befolyásolják a korábbi következtetések érvényességét. Az ilyen következtető automaták esetében a következtetés folyamatában nem lehet a korábbi következtetések eredményeit hatálytalanítani. A tervezésben viszont — gondoljunk csak a működési folyamatok mozzanataira — a következtetések eredményei általában csak egy megfelelő időtartamra vonatkoznak, vagy időponthoz kötődnek. A döntések időtől való függősége miatt szükség van temporális logika alkalmazására.

Tekintve, hogy a tervezési feladatoknak egyidejűleg több egyenértékű megoldása lehet, a modális logika szemléletének is érvényesülnie kell. Mivel döntések bizonytalan vagy töredék információk alapján is születhetnek, az eredmények nem abszolút megbízhatóságúak, sejtések és ráérzések egyaránt szerepet kapnak, a bizonytalan logika eszköztárára is szükség van. Ez összefüggésben áll azzal, hogy a tervezési folyamat sem mindig teljesen logikus vagy következetes.

A tervezés során a korábban elfogadott vagy meghatározott tények a következtetés későbbi fázisában ismertté váló tények miatt megváltozhatnak, és procedurális következetlenséget okozhatnak.

Szakértőrendszer-vázak

A fentiekben említett tények egyértelműen arra utalnak, hogy a tervező szakértőrendszereknek az alkalmazott következtetési paradigmán belül szerves egységben kell érvényesíteniük a nem monoton logika különböző formáit. Napjainkban szakértőrendszer-fejlesztést általában vagy magasszintű programozási nyelven, vagy (még gyakrabban) szakértőrendszer-vázra alapozva végeznek. A kereskedelmi forgalomban beszerezhető szakértőrendszer-vázak, amelyeknek egy reprezentatív csoportját a táblázat mutatja, tudásszemléltetési formák: például szabályokat, kereteket, szemantikus hálókat, objektumorientált és eljárásorientált struktúrákat egyaránt magukba foglalnak.

A tervező szakértőrendszereknek össze kell kapcsolódniuk grafikus, geometriai modellező és adatbáziskezelő környezetekkel is, hogy a tervezési feladatok szélesebb körét támogathassák. Az integráció a szimbolikus tudásszemléltetés és a faktuális/procedurális szemléltetés között azonban még teljesen nem megoldott. Sajnos ezekben az egyébként sokat tudó szoftvereszközökben sem sikerült még mindent megvalósítani, amit az általános tervezési feladatok igényelnének.

Horváth Imre

A NEM KERESKEDELMI CÉLÚ EGYÉNI HIRDETÉSEK KÖZLÉSE INGYENES

A kereskedelmi célú apróhirdetések tarifája gépelt soronként (azaz 60 karakterenként) 300 forint. A terjedelem alapján így kiszámított összeget kérjük az Új Alaplap Kiadói Kft számlájára átutalni (Agrobank, 219-93789 / 10878060-70030011), vagy postautalványon a kiadó címére elküldeni (1538 Budapest, Pf. 571), és feltüntetni, hogy „Új Alaplap, apróhirdetés". A befizetést igazoló szelvény másolatát — a hirdetési szöveggel együtt — a szerkesztőséghez (a kiadóéval azonos címre) küldjék el.

A szerzői jogokat sértő szoftverhirdetéseket nem tesszük közzé. (Lásd erről bővebben 1994. januári számunkat.)

Objektumorientált programozás **Clipperben**: Objects 2.0. Kérésre tájékoztatót küldök. Cím: Szűcs János, 4400 Nyíregyháza, Vasvári Pál u. 37. Tel.: (42) 313-568 vagy 312-222/1382-es mellék.

Adatmentés CD-re, streamerre; winchesterről, floppyról. Ugyanitt beszerzési tanácsadást, hálózattervezést és programkészítést is vállalok. Cím: Kovács Lajos, 1031 Budapest III., Vízimolnár u. 10. IV./33.

Stúdiómban megbízhatóan, ellenőrzötten lefordítom angol, német, francia és magyar nyelvről/nyelvre műszaki és közgazdasági folyóiratok cikkeit, hardver- és szoftverleírásait. Áfás számlát állítok ki. Cím: Szász György, 1035 Budapest III., Kórház u. 25. Tel.: 168-4874.

Számítástechnikából oktatást és korrepetálást vállalok. Továbbá ugyanitt keresek konzultációs partnereket, számítógépes barátokat és barátnőket, lehetőleg olyanokat, akik IBM PC számítógéppel rendelkeznek. Kovács Gábor, 3502 Miskolc II., Pf. 83.

IBM 386SX-16 2 MB RAM, 30MB HDD, 512 KB VGA vezérlő, 1,44 MB FDD, egér + pad, 101 gombos billentyűzet garanciával 45 ezer forintért eladó. Tel.: 169-1548

Nintendo Game Boy eladó minden tartozékkal 6.000 Ft; 1 Tiny Toon kazetta hozzá 2.000 Ft. Balatonlelle, Rákóczi út 243. Tel.: (85)354-741

Ingyen több megabájt jogtiszta (shareware, freeware) PC program! Küldj egy felbélyegzett válaszborítékot a listáért. Tilly György, 1139 Budapest, Rozsnyai u. 5.

Alaplapcsere, memória-, winchester- és floppyővítés a helyszínen. MegaSoft. T.: 295-5085

Egyszeres CD-ROM kompletten 12 000 Ft-ért eladó. Tel.: 216-7010

Vennék Soundblaster hangkártyát 3000 Ft-ig. Tel.: (72) 321-328

Keresek 386DX számítógépet színes monitorral 50 ezer forint körüli áron. Cím: Varga Zoltán, tel.: 06-36-354-218, Érd.: 17 óra után az esti órákban.

Vírusmentes PC-s programokat (főleg demókat) cserélek. Válaszboríték előny! Listát kérek! Cím: Varga Zoltán, 9731 Kőszeg, Pf. 9.

ELENDER COMPUTER

1087 Budapest, Hungária krt. 8.
Tel.: 134-5214, 114-0532 Fax: 133-4347

1134 Budapest, Csángó u. 13. Tel./Fax: 270-3097
4029 Debrecen, Piac u. 57. (Amfóra udvar) Tel./Fax: (52) 413-795
6721 Szeged, Madách u. 15. Tel./Fax: (62) 310-269
8200 Veszprém, Zrínyi u. Botev üzletház Tel./Fax: (88) 428-235
9700 Szombathely, Hunyadi u. 45. Tel./Fax: (94) 312-265

ELENDER

Nyitva: hétfőtől péntekig 9-17 óráig

Maxtor PCMCIA cserélhető winchesterek

171 MB, PCMCIA III.
Operating Shock: 120 Gs
Non-operating Shock: 600 Gs
MTBF: 300.000, 14 ms., 10x53x84 mm

MobileMax 20 MB

Flash card-ok

2 MB 4 MB 8 MB 12 MB 16 MB 20 MB

Az ötcsillagos szakvásár

1995 május 9-13.
Várjuk Önt az
"A" pavilon 212/d standján

1995 május 2-án nyitja
az ELENDER KFT.
SAMSUNG mintaboltját:
1092 Budapest, Ferenc krt.16.
Telefon/Fax: 218-2858

IFABO BUDAPEST 1995

MAXOPTIX T4
OPTIKAI DRIVE

Maxoptix

Paraméterek:
- 1.3 GB
- 34 ms hozzáférési idő
- 2.2 MB/s
- SCSI II.
- 1 MB Cache
- 41x146x203 mm

Biztonság:
- 100.000 óra MTBF
- Novell bevizsgált

INFORMÁCIÓKÉRÉS: 0518 ▶

RICH SELLING SOFTWARE

SZOFTVER MINDENKINEK !

-Számlázás, készletnyilvántartás
-Komplett pénztár-, raktárkezelés
-Gépkocsi-, bér-, ÁFA elszámolás
-Teljeskörű kimutatások
-Szövegszerkesztő, üzleti levelezés
-Egyedi listagenerátorok
-Moduláris bővíthetőség

Komplett ügyviteli rendszer
az eladási ár 1/5-éért

Regisztrációs díj 20000 Ft.

Együtt, egy rendszerben

RICH SELLING KFT
BUDAPEST 1135
PETNEHÁZY U. 75/A.
Tel.: 1-402-404
Fax: 1-490-953

INFORMÁCIÓKÉRÉS: 0555 ◀

Pro/ENGINEER

A csúcskategóriájú 3D-s CAD/CAM rendszer.

Piacvezető világszerte.
Magyarországon közel 70%-os piaci részesedés.

Hogy a csúcstechnológia elérhetőbb legyen,
a Parametric Technology Corporation bejelentette a

Pro/JR.

3D-s CAD csomagot.

Jellemzők röviden:
parametrikus 3D-s testmodellezés
intelligens alaksajátosságok
hatékony összeállítás kezelés
automatikus rajzkészítés
kétirányú asszociativitás
DXF, IGES,SET, RENDER, stb. interfészek
valós idejű takart vonalas, árnyékolt mozgatás

CREATIVE

CREATIVE Engineering Kft.

Bemutatóterem : Budaörs, Fodros utca 47/b
Tel.: 276-3701, 277-9359 Fax: 274-2094

INFORMÁCIÓKÉRÉS: 0513 ▶

**Nyáktervezés,
mesterfilm-készítés
bármely GBR-formátumból,
valamint SMARTwork fájlból,
nyákgyártás.**

**Szövegfilm-levilágítás
PC és Macintosh rendszerről,
szkennelés, színrebontás,
kromalinkészítés.**

FILM NYÁK BT.

**1082 BUDAPEST,
LEONARDO U. 50.**

**Telefon,
fax,
modem:
134-2600**

INFORMÁCIÓKÉRÉS: 0542 ◀

Ikonológia

Képtelen képességek köntöse

Amikor a programozás iránt érdeklődő olvasóink az elmúlt években a Snobol nyelvről szóló cikkekkel találkoztak lapunkban, bizonyára eszükbe ötlött, hogy ezeket a frappáns megoldásokat nem lehetne-e kevésbé extravagáns formátumú nyelvben alkalmazni. Vajon szükségszerű-e, hogy a különleges módszerek alkalmazásához úgy kelljen beleszoknunk egy új nyelv gondolkodásmódjába, hogy a jelölésmód faramuciságait is kénytelenek legyünk elviselni? Természetesen nem szükségszerű.

Az USA-ban még a mikroprocesszor-korszak hajnalán elindult egy kutatási projekt, azzal a céllal, hogy mentse át egy külső megjelenését tekintve „klasszikus" programnyelvbe a Snobol eredeti gondolatait, és az új formában keresse a továbbfejlesztés lehetőségeit. Nos, ekkoriban kezdett kialakulni a fejekben az Icon nyelv koncepciója. A régi nyelv tovább élt a régi nevén, a születendő nyelvnek azonban más nevet illett választani.

Nyolc és fél

Eleinte nagygépes környezetben dolgoztak a fejlesztők, és eszük ágában sem volt mikroszámítógépre váltani. A számítógépek fejlődése azonban felrúgta eredeti elgondolásaikat. A nagygépek után először VAX gépeken jelentek meg az újabb Icon-verziók, majd rövidesen a PC-s korszak is elkövetkezett. Egy idő múlva már a nyelvfejlesztő gárdának kellett kapaszkodnia, hogy lépést tartson a számítógépes fejlődéssel.

A verziószámok gyors növekedése nem annyira a belső fejlődés igényéből fakadt, mint inkább a külső körülmények folyamatos változásából. Ahhoz, hogy kellően hasznosítani tudják a DOS 386/486 „igazi" 32 bites védett módú platformjának lehetőségeit, már az Icon 8.0 verziójának tudása is kevésnek bizonyult; ki kellett fejleszteniük — a Fellinit idéző — 8.5 verziószámú változatot, amely a koprocesszort is segítségül hívja a számításigényes feladatok megoldásához, és csak kénytelenségből fanyalodik az emulációra. Azóta egyébként bizonyára készül, vagy már el is készült az újabb verzió az újabb processzorok illő fogadására.

Az újabb verzióknak bizonyos mértékig sikerült csökkenteniük azt az elszigeteltséget, amely a Snobol nyelvet távol tartotta a számítástechnikai fejlődés fő áramlatától. A Snobol nyelven írott programok magas fokú intelligenciáját más nyelvekből csak nehézkesen — fájlokon keresztül — lehetett hasznosítani, illetve más nyelveken készült eljárásokat nem lehetett Snobol programokból futtatni. Az Icon ma már nemcsak formailag hasonult a klasszikus programnyelvekhez, hanem bizonyos mértékig sikerült belesimulnia C nyelvű környezetébe: Icon programokból lehet hívni C nyelven írt függvényeket, és az Icon programok is aktivizálhatók C nyelvű programokból. (Egyébként maga az implementáció is C nyelven készült.)

Aki a Windows ikonjait keresi ebben a régi-új nyelvben, annak alighanem csalódnia kell. Elkészítették ugyan már az Icon nyelv „ablakos" változatát, de csak X-Windowra. Semmi elvi vagy gyakorlati akadálya nincs annak, hogy mutogatással lehessen kiválasztani a nyelvi eszközöket. Az egérrel való mutogatás, bár nem ellenkezik a nyelv szellemével, semmiképpen nem tartozik a lényegéhez. Vagyis előbb lett ikon az Icon, mintsem maga a szó gyökeret vert volna a számítástechnikában (a ma használatos leszűkített értelemben).

Lássunk egy medvét!

Legjobb, ha egy apró programrészlettel mutatjuk meg az Icon néhány fontos jellemzőjét.

```
while sor := read() do
    write(sor)
```

Ez a két sor nem egyetlen sort olvas be és ír ki, hanem mindaddig működik, amíg el nem fogy az olvasnivalója, akkor azonban kérés nélkül leáll. Az Iconban szétválasztottak két dolgot: az utasítás eredményét és a végrehajtás sikerességét. A program folytatását, a vezérlést nem az eredmény értéke irányítja, hanem a vezérlés sikeressége. A sikertelenség nem valamilyen logikai „false" eredményt produkál, hanem semmilyen eredményt. Ha a beolvasás előző eredményét egy változónak adtuk át, akkor az a változó sikertelenség esetén nem kap új értéket, hanem tovább őrzi a régit.

Ha viszont a fenti összetett utasítás sikeresnek bizonyul, akkor a while-do páros utasítás tovább kell, hogy működjön. Akkor sem történik semmi baj, ha elfogy az input. Éppen csak véget ér az utasítás végrehajtása, és rátérhet a soron következőre (amit a fenti részlet után odaképzelhetünk).

A „sor" változó értéke attól függ, hogy egyáltalán volt-e a read() függvénynek olvasnivalója. Ha nem, akkor megtartja azt az értékét, amelyet a fenti programrészletre térése előtt felvett. (Ha első előfordulása a változónak, akkor kezdeti értéke az „üres" érték.) Ha viszont volt már mit beolvasnia, akkor az utoljára beolvasott sor értékét „őrzi féltve" a változó, sikertelen kísérlet után is.

Még rövidebben

A fenti két sort még tömörebbé is tehetjük a következő jelöléssel:

```
while write(read())
```

Látható, hogy a do opcionálisan használandó része a while-do párnak, ha nincs rá szükség, elhagyhatjuk. Ha viszont több tennivalót is szánunk a do után következő

utasításnak, akkor a több utasításból álló egységet a C nyelvből ismert jelöléssel „bajuszos zárójelbe" kell tennünk.

A fenti kis egysoroson még egy fontos dolgot megfigyelhetünk. Előfordulhat, hogy egy függvény kiértékelése során már az argumentum kiértékelése sikertelen lesz. Ekkor a sikertelenség „öröklődik", tehát az argumentumot tartalmazó függvényre száll át a sikertelenség. Ha történetesen több függvényhívás volt egymásba ágyazva, akkor ez az örökség automatikusan továbbadódik, egészen addig, míg a legkülső függvényt el nem éri. Természetesen mindez anélkül, hogy ilyenkor a program egyáltalán belefogna a sikertelenség hírét továbbító függvények kiértékelésébe.

Ez a vezérlési mechanizmus jól kihasználható, bár veszélyeit is érdemes észrevenni: elsikkadhat az az információ, hogy hol lépett fel először a sikertelenség. Persze vannak azért más módszerek is arra, hogy el lehessen kapni az elveszettnek hitt információ frakkját...

Teve van egypupú...

A fenti medvebocs körmöcskéinek megismerése után lássunk egy bonyolultabb példát. Először leírunk egy több részből álló összetett utasítást, majd utána megnézzük, melyik része mit csinál.

```
every (i := upto(szo2, szo1) & (j := upto(szo1[i], szo2)) do {
 every write(right(szo2[1 to j - 1], i))
 write(szo1)
 every write(right(szo2[j + 1 to *szo2], i))
 write()
}
```

Kezdjük belülről a kifejezések megértését. Az

```
1 to j - 1
```

kifejezés előállítja a számokat egyesével, 1-től j – 1-ig, ezt könnyű kitalálni.

A szo2 előtt álló csillag azt jelenti, hogy a szo2 változóban őrzött sztring hosszát kell venni. A másik hasonló kifejezés tehát előállítja a számokat j + 1-től a szo2 sztring hosszának megfelelő számig.

A szo1[i] a szo1 sztring i-edik elemét jelenti, ahol az i helyén bonyolult kifejezés állhat. Így tehát az előbb előállított 1, 2, 3, ... j – 1 sorozat elemei is, csak nem egyszerre, hanem egymás után.

Egy fifikásabb függvény

Ravaszabb az upto( ) függvény működése. Ez a függvény az első argumentumaként megadandó karakterhalmaz elemeit veszi sorra, és elkezdi hasonlítgatni a második argumentumként megadandó sztring karaktereivel. Ez a hasonlítgatás addig tart megállás nélkül, amíg egyezést nem talál. Ekkor büszkén leáll, és felmutatja a trófeát: az első olyan karakter pozíciószámát a sztringben, amely megfelel a követelményeknek. Sikertelen is lehet a függvény működése: ha a sztring egyetlen karaktere sem felel meg a karakterhalmaz egyetlen elemének sem.

Igen ám, de szépen fejlett utasításunkban mindkét esetben az upto függvény első argumentumának helyén nem karakterhalmaz van megnevezve, hanem egy sztring. No problem. Automatikusan végbemegy az átalakítás, tehát ha például az

„Alaplap" lenne szo1 és szo2 közül valamelyik, akkor a sztringből automatikusan kialakulna az 'Aalp' rendezett karakterhalmaz. (A rendezettséget az ASCII karakterkódok értékei határozzák meg.) Az ilyen konverziókat kéretlenül is elvégzi a program, nem áll le hibajelzéssel, mint egy pedáns Pascal program.

Kezd már világosodni...

Kezdjük már érteni az utasítás első sorának értelmét, legalábbis azt, ami az every után áll:

Az i értéke annak a karakternek a pozíciószáma lesz, amelyet a szo2-ből alkotott karakterhalmaz első olyan karaktere alapján állapítunk meg, amely legelőször található meg (balról jobbra haladva) a szo1 sztringben.

Ezt az i értéket választjuk indexnek a szo1 sztring megfelelő karakterének kiválasztására. Figyelem! Ez a kiválasztott egyetlen karakter egy egyetlen elemű karakterhalmazt alkothat, különösebb konverziós erőlködésre tehát itt nincs is szükség. Csupán ezt az egyetlen karaktert kell megkeresni a szo2 sztringben, és ennek pozíciószámát beírni a j változóba.

A right(s,i) függvénynek nagyon egyszerű a szerepe: az s-sel jelzett sztringet i betűhellyel jobbra csúsztatja, és balról „kipárnázza" szóközökkel, vagy ha kell, egy harmadik argumentumként megadott karakterrel.

Ideje azonban, hogy az every-do „páros kötőszó" jelentését (és jelentőségét) is felfogjuk.

For ciklus — másképp

Mit ad vajon eredményül az alábbi kifejezés?

```
every i := 1 to 10 do
  write(i^2)
```

Nem nehéz kitalálni, hiszen már ismerjük az 1 to 10 részkifejezés szerepét: sorra elő kell állítania az 1...10 számokat. Ezeket az értékeket veszi itt fel az i változó. Miután az i értéket kapott, a do „kötőszó" után következő függvény argumentuma is értéket kap, s a parancsnak megfelelően megjelennek az outputon az i érték négyzetének megfelelő számok. Ahogyan tehát a „1 to 10" kifejezés sorban adogatja az i értékeket, úgy generálja az every-do „kötőszópár" az összetett utasítás alapján kiszámított értékeket.

Nicsak: mintha részeire trancsíroztuk volna a más nyelvekből ismert „for ciklust". Vajon miért? Egy másik példából ez világosabban kitűnik. Hasonlítsuk össze az A) és B) kifejezést:

```
A) write(szoveg ? find("the"))
B) every write(szoveg ? find("the"))
```

Mit fog kiírni az első kifejezés? Amit a write() függvény argumentuma előír: megkeresi a „the" szócska első előfordulásának kezdő pozícióját egy sztringben, amelynek neve a kérdőjel-operátor előtt látható azonosító. Magát a keresést a find() függvény nevezi meg, a kérdőjel-operátor azt jelenti, hogy az előtte álló operandust valamilyen célból „végig kell nyálazni".

Ha most elébe írjuk az egésznek az every szót, mint ez a B) kifejezésben látható, akkor a kiértékelés ciklusban hajtódik végre. Nem ragad le tehát a végrehajtás az első előfordulás pozíciójának meghatározásánál, hanem sorban generálja az összes többi előfordulás pozícióját is. Hasonló a helyzet, mint a while-do „páros kötőszó" alkalmazásakor: addig tart a ciklikus végrehajtás, ameddig sikeresen folytatható. Az analógia annál is inkább helyénvaló, mert itt ugyanúgy elmaradhat az opcionális „do rész", mint ott.

Most már érthető az egész!

Ennyi tudással a tarsolyunkban térjünk vissza jól fejlett utasításunkhoz. Tekintsünk el most a bajuszos zárójelen kívül lévő résztől, csak az ezen belül előírt tevékenységek érdekeljenek.

Amíg a write(right(szo2[1 to j - 1], i)) kifejezéssel jelölt ciklus „le nem jár", addig i hellyel el kell tolni a szo2 karaktersorozatból kiemelt első, második, ... j – 1-edik karaktert. Figyeljük meg: egyetlen sorban egyetlen karaktert. Önmagában ez a rész függőlegesen kiírja a szo2 sztring j-edik karaktere előtt álló karaktereket, mindegyiket az i-edik pozícióba.

Ezután kiírjuk vízszintesen a szo1 sztringet, majd folytatjuk a szo2 szó függőleges kiírását a j + 1-edik pozíciótól.

De mit tesz isten? Az első sor úgy válogatja össze az i és j értékeket, hogy a vízszintesen és függőlegesen leírt szavak éppen közös elemükben találkozzanak, abban, amelynek helyét a szo2-nek j-edik, a szo1-nek pedig i-edik pozíciója határozza meg. (Egyúttal ez az a pozíció, amelyben a függőlegesen írt betűk is elhelyezkednek.) Közben persze az every azt is biztosítja, hogy végül minden olyan „szókereszt" ki legyen írva, amelyben a két szónak van közös eleme.

Hogy lehet tetszőlegesen megadott szo1 és szo2 szavakon végrehajtatni ezt a tevékenységet? Például úgy, hogy a megadott utasítást kikerekítjük eljárássá:

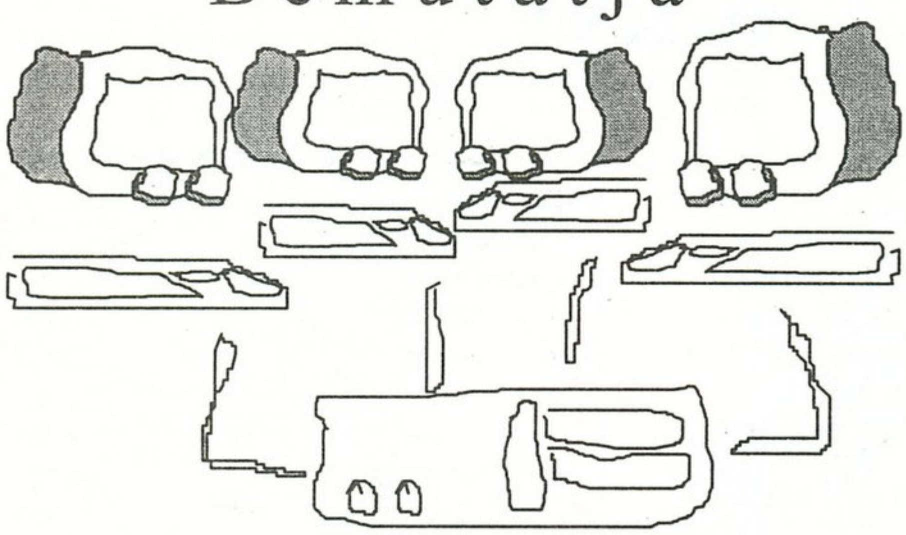
```
procedure keresztez(szo1, szo2)
 local i, j
 ...
 (ide jön a fenti kifejezés)
 ...
end
```

Elképesztő teljesítmény — lényegében egyetlen utasítással felírva!

Vargha Dénes

Lightstone család

Bemutatja



CBI Computer Business Info Kft.

7030Paks Dózsa Gy u. 51-53.Tel.:75-312-861

A TAKARÉKOS MEGOLDÁS!

INFORMÁCIÓKÉRÉS: 0507 ▶

Keresse könyveinket!

COMPUTERBOOKS

Dr.Kovácsné C.J.-Ozsváth M.:
Windows for Workgroups 3.11
- hálózattal vagy anélkül 1.115.-
Benkő T.né-Benkő L.: **MS-WORKS 3.0 a mindennapi életben** - magyar verzió 793.-
Benkő T.né-Kuzmina J.-Kiss Z.-Dr.Tamás P.-Tóth B.: **Könnyű a Windows-t programozni!?** átdolgozott kiadás - lemezzel 1.683.-
Nagy G.: **Kézikönyv az adattömörítéshez - ARJ, PKZIP, & Co.** - lemezzel 1.298.-
Borgulya I.: **Szakértői rendszerek, technikák és alkalmazások** 1.375.-
Dr.Kovácsné C.J.-Takács T.:
Ismerkedés az SSADM-mel 966.-
Abonyi Zsolt:
PC hardver kézikönyv
bővített, átdolgozott kiadás 875.-
Dr.Kovácsné C.J.-Benkő L.-Dr.Pergel J.-né:
Mindenkinek! a PC-ről - DOS, Windows felhasználóknak - átdolgozott kiadás 499.-
Kovalcsik Géza:
Excel for Windows 5.0 magyar * angol 1.147.-

Kérje katalógusunkat!

Levélcím:
1253 Budapest Pf.: 71.
Bp.,XII.Tartsay V.u.12.
Tel.: 175-1564
Tel./fax: 175-3591

INFORMÁCIÓKÉRÉS: 0510 ▶

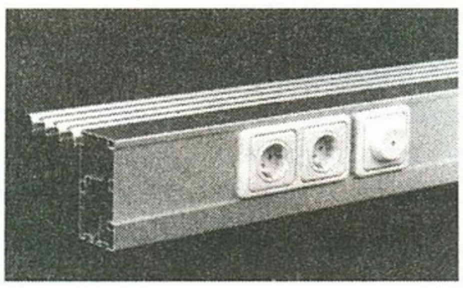
ELECTRAPLAN KFT.

Adatátviteli kábelhálózatok ideális szerelési rendszere az

ELECTRAPLAN

padló alatti és mellvédszerelési rendszer

Licenc: **D.E.G.** Hamburg



- **Mellvédcsatornák** alumíniumból, műanyagból, acéllemezből
- **Energiaoszlopok** alumíniumból

- **Padlócsatlakozók** már 5 cm-es aljzattól bármely burkolathoz 16 szerelvényig
- **Padlócsatornák zárt** és **nyitható kivitelben bármilyen burkolathoz** elektromos, víz-, fűtés- és egyéb technológiai cső vezetésére.



- A különféle adatátviteli csatlakozók beépíthetők készülékeinkbe
- Kiszolgálás **raktárról**
- **Pótalkatrész**-ellátás
- **Egyedi gyártás** is

Gyártó és forgalmazó:

ELECTRAPLAN KFT.

1134 Budapest, Lehel u. 23. Tel/fax: 129-8393; 270-3775

INFORMÁCIÓKÉRÉS: 0517 ▶

Rezon TRADE

1065 Budapest, Nagymező u. 40.
Tel: 132-0553, 132-4572, 132-8906 Fax: 111-9277

Német számítógépes ***öntapadó címkék*** *többféle színben lézer- és tintasugaras nyomtatókhoz, valamint fénymásoló gépekhez, A/4 íven is.*

Osztrák számítógépes ***leporellók*** *1–5 példányig raktárról.*
25 éves olvashatósági garancia!

Számítógépes egyedi nyomtatványok tervezése, a nyugat-európai standardnek megfelelő színvonalas gyártása Ausztria legfejlettebb nyomdájában.

Színes, 80 g-os ***Dinocolor fénymásoló papír****, lézer- és tintasugaras nyomtatásra is kiválóan alkalmas. 25–500 íves kiszerelések szivárvány, színátfutó és intenzív színekben.*

Cégek teljeskörű író- és irodaszer-ellátása házhoz szállítással.

Megrendelését telefonon és telefaxon is elfogadjuk, a kívánt árut 20 000 Ft értékhatár felett Budapest területén ingyenesen házhoz szállítjuk.

INFORMÁCIÓKÉRÉS: 0554 ▶

„Pascal-trilógia”

Három szint — három könyv

A ComputerBooks gazdag kínálattal traktálja a számítástechnikai ismeretekre vágyó olvasókat. Meglepő módon néha elébe is megy a szoftverpiac kínálatának, de lemaradni semmiképpen nem akar tőle. Amikor kiadványaik erényei mellett azok hibáira is rámutatunk, nem a kiadó (háromfős) stábját akarjuk piszkálni, és nem is a szerzők érzékenységét akarjuk sérteni. Egyszerűen szeretnénk, ha magasabbra emelnék a mércét ott is, ahol időnként lankad a figyelmük. Legközelebb így talán sokkal kevesebb kritizálnivalót fognak kiadványaikban felejteni... Most a „Pascal-trilógiát” vettük szemügyre.

Didaktikailag ügyes felépítésben sok hasznos információt tartalmaz már a trilógia első kötete is. Frappáns példaprogramjainak megértését nagyban megkönnyíti a lemezmelléklet, amelyen megtalálhatók (és Pascal-fordítóval kipróbálhatók!) a szöveg közben előforduló forrásnyelvi programok.

A „Pascal-trilógiának” a második a legjobban kidolgozott és talán a legértékesebb darabja, bár — meglepő módon — éppen ez jelent meg a legszerényebb külsőben.

Koncepciójában is, megvalósításában is kitűnő, rendkívül hasznos művet alkottak a szerzők. Több szerzőről lévén szó, szemléletük árnyalatokban bizonyára eltér egymástól, de a köteten mindez nem érződik, viszont az összehangolás nyelvi megvalósítása hagy némi kívánnivalót maga után.

Információmennyiségben a harmadik kötet tartalmazza a legtöbbet. Ez bizonyos mértékig érthető is, hiszen a legnehezebb területeken kíván segítséget nyújtani: a DPMI megismerésében és használatában, valamint a Windows alatti programozás műhelytitkainak elsajátításában.

A DPMI rövidítés (DOS Protected Mode Interface) a DOS védett módú alkalmazásának szabványos, hardverfüggetlen felületére utal, amelyet 1990-ben dolgoztak ki és fogadtak el a világ vezető szoftverfejlesztő cégei. Ez annyi mindent foglal magában — leírótáblák kezelése, üzemmódok közötti váltás, a bővített memória elérése, DOS-módszerű memóriafoglalás, a megszakítási rendszer vezérlése, a valós módú programokkal való kommunikáció —, hogy mindezen tevékenységek végrehajtásához külön kiszolgáló rendszert kellett kifejleszteni.

Itt jegyezzük meg, hogy a Turbo Pascal 7.0, amelyet a cég együtt ad a Borland Pascal 7.0-val, nem szerves része a BP 7.0 rendszernek, hiszen valós módban működik, csak a hagyományos 640 kbájt memóriával gazdálkodik, így tehát nincs is szüksége erősebb eszközökre. Ennek ismertetése nem is tartozik bele a könyv témakörébe.

Benkő Tiborné—Benkő László —Tóth Bertalan—Varga Balázs:

Programozzunk Turbo Pascal nyelven

Harmadik, javított kiadás. ComputerBooks, 1995. 500 oldal, 796 Ft (lemezmelléklettel).

A fejezetek kisebb tematikus egységei szerint csoportosítva kellemesen sok gyakorlatot talál az is, aki nemcsak a Pascalban, hanem a számítástechnikában is kezdő. Biztonságérzetét növelheti, hogy a legtöbb gyakorló feladat megoldása megtalálható a lemezmellékleten. (A Turbo Pascal arra is kiváló eszköz, hogy ki lehessen próbálni: mi lenne, ha itt vagy ott valamit változtatna a programon!)

Külön fejezetek foglalkoznak a TP fájlkezelésével és memóriahasználatával, valamint a CRT unit és a GRAPH unit részletes ismertetésével. Valamivel elnagyoltabb a megszakítások és a tárrezidens programok használati módjá-



nak és használati lehetőségeinek bemutatása.

Bizonyára sok olvasónak okoz örömöt a könyv azzal, hogy kitüntetett figyelmet (sőt önálló fejezetet is) szentel különböző játékprogramok írásának, elemzésének. A bemutatott játékok példáján bevezeti az olvasót a programozás rejtelmeibe, a játék logikájának kihámozásától kezdve a működéshez szükséges eljárások és függvények kidolgozásán keresztül a teljes program elkészítéséig.

A függelékek tartalmazzák a TP szabványos eljárásait és függvényeit, a CRT és a GRAPH unit eljárásainak és függvényeinek ismertetését, a fordító direktívákat, valamint a futás közben jelentkező hibakódok rövid értelmezését.

Kár, hogy a szöveg fogalmazása helyenként pongyola, pontatlan (pl. van, ahol normál alakúnak nevez nem normál alakú számokat, hibásan használja a helyiérték-elnevezést stb.). Néha alighanem a sok szerző között vész el az egységes terminológia, ami különösen a kezdő programozók számára lehet zavaró. (Nem is mindig szerencsés a

szóhasználat: pl. hibás asszociációhoz is vezethet külső megjelenési formája alapján idézetnek nevezni azt, amit másutt karakterláncnak hív.)

A fő kifogás azonban a szöveg helyesírási pongyolaságát illeti. Szinte érthetetlen, hogy a szerzőknek, akik egy formális nyelv szigorú szintaxisát magyarázzák, ilyen kevés érzékük legyen a magyar nyelv szintaxisát tükröző központozási szabályok iránt. Aki a számítástechnikában ismeri a típus fogalmát, annak nem szabadna összekevernie a magyarban sem, hogy melyik „mint" szó elé kell vesszőt tenni, és melyik elé nem. Aki a programok szerkezeti tagolására gondosan ügyel, annak nem szabadna a mellékmondatokat sem lezáratlanul hagynia. A „ha"... „akkor"... szerkezetek hibás helyesírása és a magyar mondatszerkesztés „szintezési szabályainak" egyéb megsértése olyan gyakori jelenség a könyvben, hogy azt nehéz lenne egyszerű figyelmetlenségnek minősíteni.

A könyv értékét nagyban emelné, ha a jelenleginél **sokkal részletesebb** lenne a tárgymutatója. Ennek többszerzős mű esetén különösen nagy jelentősége van, már csak a terminológia egységesítése szempontjából is. De „mellékhatásként" a szerzőknek is sok helyen segítséget nyújtana az egybeírás—különírás és a kötőjeles írás kényes problémájának a megoldásához. (Nem fordulhatna elő például, hogy akár egyetlen bekezdésen belül is többféleképpen írják ugyanazt a kifejezést.)

Örömmel vennénk, ha a negyedik kiadás már ezeknek a zavaró hibáknak a kijavításával készülne el. A könyv egyébként igazán értékes tartalma megérdemelné a szöveggondozás minőségileg magasabb szintjét.

Benkő Tiborné—Benkő László—Kiss Zoltán—Tóth Bertalan:

Objektum-orientált programozás Turbo Pascal 6.0-ban, (Turbo Vision)

ComputerBooks, 1994. 312 oldal, 636 Ft (lemezmellékletttel).

A könyv két nagy részre oszlik. Az elsőben részletesen és szemléletesen bemutatják a szerzők a TP 6.0 teljesen új integrált fejlesztői környezetét (Integrated Development System, IDE), majd összefogottan, a kellő mélységig leírják az objektumorientált programozás elméletét és gyakorlatát. Ezután a TP 6.0 használata következik az installálástól kezdve a beépített assembler tömör, lényegre törő ismertetéséig. A könyv második (nagyobbik) része a Turbo Vision keretrendszerrel foglalkozik. A szerzők lépésről lépésre vezetik be az olvasót ennek a keretrendszernek az értő használatába, meghagyva neki a felfedezés jóleső örömét is az újszerű dolgok megismerésében.

A Turbo Vision egyidejűleg eszköze



és tárgya is az objektumorientált programozás megismerésének két fontos területen: az ablaktechnika támogatásában és az eseményvezérelt programok készítésének elsajátításában.

Lényeges újítása a TP 6.0 új integrált környezetének, hogy egyszerre több, egymást többszörösen átfedő ablakot is kezelni tud — éppen a Turbo Vision által biztosított lehetőségek kihasználásával. Ha nem is a Windows grafikus képernyőkezelésének szintjén, de lehetővé válik különböző tulajdonságú ablakok létrehozása, mozgatása, méretének megváltoztatása stb. A számítógéppel való kommunikáció biztosításához könnyűszerrel tudunk ebben a környezetben különböző menüket, dobozokat készíteni az ablakozás módszerével. Sokan vannak, akik a Turbo Pascal 6.0-s változatát anélkül használják, hogy igazán tudatában lennének a benne szunnyadó lehetőségeknek. Ez a könyv most kitűnő alkalmat nyújt arra, hogy továbbfejlesszék TP-s ismereteiket, és képesek legyenek igazán profi megjelenésű programokat vagy programrendszereket kifejleszteni.

Igen részletesen és alaposan foglalkozik a könyv az események fogalmával, különböző fajtáival, és azzal, hogy miként használhatók fel az események a programok futásának vezérlésére. Aki mindezekkel a fogalmakkal grafikus környezetben találkozik először, könnyen elveszhet a technikai részletekben — itt viszont alkalma van, hogy mélyebben megértse a végbemenő folyamatokat. Az „egér-események", „billentyűzet-események", „üzenet-események", „üres események" szerepének tudatosítása egészen más szemléletet kíván a programírásban is, mint amihez előzőleg hozzászokhatott a lineáris felépítésű programok sokkal egyszerűbb világában.

Akik az újabb szoftvereszközök farkasétvágyát megismerték, azokban időnként óhatatlanul felmerül a kérdés: meddig lehet még bírni szusszal (és más egyébbel) ezt a megalomániát. Néha nyugvópontokat is kellene keresni (ha csak ideiglenesen is) az operációs rendszerben és kedvenc fordítóprogramunkban olyan verziónál, amelyek legalább egy-két évre kielégítik az igényeinket. Aki a DOS 3.3 után azonnal áttért a 4.x verzióra, az utána jobban meggondolja, hogy mikor, minek a kedvéért hagyja ott a TP 6.0-t. Valóban, a Turbo Pascal 6.0-s változata olyan kiforrott termék az integrált fordítói környezetek között, amelynél lehet egy kis pihenőt tartani. Nem arról van szó, hogy a TP 7.0 ne szárnyalhatná túl, de nagyobb teljesítményű eszközre áttérni csak annak éri meg, aki annak megfelelő alkalmazások elkészítésébe vág bele, és már kacsingat a védett üzemmód lehetőségei felé.

Aki most fontolgatja az elmélyülést a tárgyban, mert csábítják a Windows grafikus felületei, az előtte még gondolja át, hogy szerzett-e már kellő tapasztalatokat az objektumorientált programozásban — egyszerűbb körülmények között. Nem mindegy, hogy melyik az a fejlődési fok, amelyet át akarunk ugrani. Aki nem akarja, hogy számos kudarcélmény árán (és feltehetőleg sokkal lassabban!) jusson csak el az óhajtott célig, az jobban teszi, ha nem sajnálja az energiát a szöveges képernyős fejlesztés megismerésére. Nos, a Turbo Visionban a szóban forgó könyv kalauzolásával olyan igényes keretrendszert ismerhet meg, amelyet maga a Borland is egyik legfontosabb eszközként használt rendszereinek kifejlesztésében. Megismerése így azoknak is hasznos lehet, akik most éppen várakozó álláspontra helyezkedtek, és a szellemi muníciót gyűjtik a továbblépéshez.

Érdemes külön kiemelni a könyv gazdag példaanyagát: jó segítséget nyújt az elméletben megismert fogalmak manipulatív megértéséhez és felhasználásához. Fontos szerepe van a lemezmellékletnek: érdekes és jól kidolgozott programjain keresztül biztosítani tudja az olvasó aktív részvételét az anyag elsajátításában.

Benkő Tiborné—Kiss Zoltán—Tamás Péter—Tóth Bertalan:

Programozás Borland Pascal 7.0 rendszerben (DPMI, Windows)

ComputerBooks, 1994. 660 oldal, 1586 Ft (lemezmelléklettel).

A Borland Pascal DPMI szervere — a Windows 3.x szerveréhez hasonlóan — a hagyományos DOS- és BIOS-rendszerhívásokat is támogatja, bár maga az alapul vett szabvány (legalábbis a 0.9 verzió) ezt még nem írja elő. Abban is megegyezik a Borland megoldása a Windows 3.x alkalmazásokkal (sőt az OS/2 1.x alatt futó alkalmazásokkal is), hogy az RTM futtatórendszer (Run Time Manager) ugyanolyan futtatható fájlformátumot használ a végrehajtható .EXE programokra vonatkozóan ugyanúgy, mint a dinamikusan szerkeszthető .DLL könyvtárak (Dynamic Link Library) betöltésénél. Maga a DPMI szerver egy overlay fájlban van elhelyezve (DPMI16BI.OVL), amely 16 Mbájtig a gép teljes bővített memóriáját ki tudja használni.

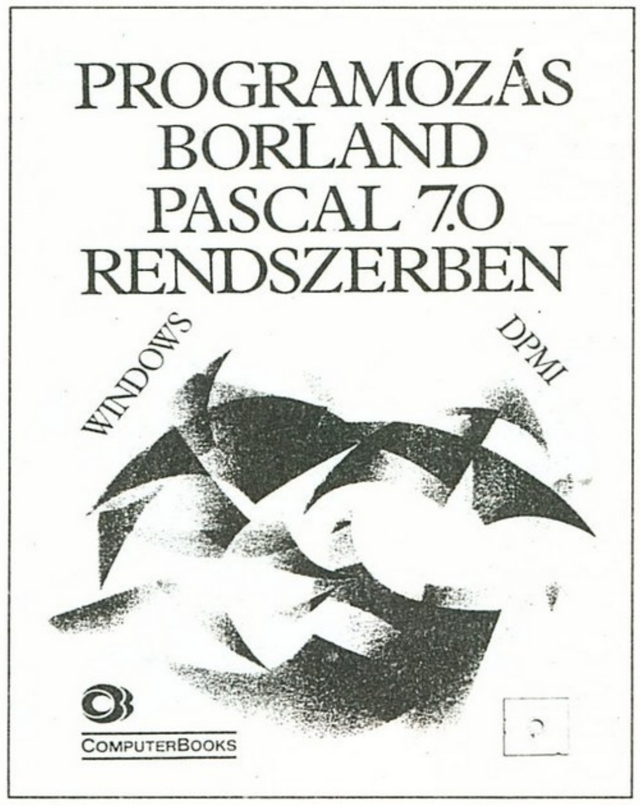
A .DLL könyvtárak használatának az a legfőbb előnye a Borland cég hagyományos Pascal unitjaival szemben, hogy kódjai megosztva, több programból is használhatók. A Windows alatt gyakori jelenség, hogy ugyanannak a programnak több példánya fut. Ilyenkor az RTM biztosítja a többszörös hozzáférést, azt tehát, hogy a megfelelő .DLL-nek csak egyetlen példányára legyen szükség a memóriában.

A .DLL-koncepció Borland-fejlesztésének további újdonsága is van a Windows 3.0 és 3.1 megoldásaival szemben. Nevezetesen az, hogy a Windows hagyományos API (Application Programming Interface) felületének szokásos moduljain kívül (WinTypes, WinProcs és Win31) a Borland kifejlesztett egy ún. WinAPI modult is annak biztosítására, hogy a .DLL könyvtárak egyaránt használhatók legyenek a Windowsból és a DOS védett módjában futó programokból is. A Borland DPMI bővítése egy olyan védett módú memóriakezelőt tartalmaz, amely működésében megfelel a Windows 3.x rendszerek globális memóriaelérését vezérlő menedzser-tevékenységének, rutinjainak hívása pedig binárisan kompatibilis a Windows API hívásaival. A WinAPI hívásait egyébként a BP 7.0-ban ugyancsak az RTM dolgozza fel. Mivel azonban a hívások formailag (paraméterezésükben) megegyeznek a Windows API hívásaival, olyan .DLL könyvtárak is szerkeszthetők, amelyek a Windows rendszerrel közösen használhatók. Bizonyos Windows-erőforrások közvetlenül beleszerkeszthetők BP programokba, sőt arra is van lehetőség, hogy C-ben készült .DLL-eket DPMI üzemmódban használjunk BP 7.0-ból.

Ezeket a lehetőségeket — technikai megvalósításukkal együtt — nagyon szépen, világosan kifejti a könyv első egynegyed részében. Mihelyt azonban a Windows alatti programozásra kerül a sor, sokkal kevésbé átgondoltnak, didaktikailag ügyetlenebb szerkezetűnek tűnik a közölt ismeretanyag. Belevágnak a szerzők a Windows programozás alapjainak taglalásába, mielőtt részletesebben ismertetnék a Windows memóriakezelését, az eseményként érzékelt adatbevitelt (egérmozgatás, billentyűleütés stb.) és az ún. „erőforrásokat", vagyis azokat az .EXE fájlokba megváltoztathatatlan formában belesuvasztott kiegészítő adatokat, amelyeket az alkalmazói program csak szükség esetén tölt be a memóriába. Így azután a technikai részletek kásahegye sehogyan sem akar elfogyni, többszöri elolvasásra, oda-vissza keresgélésre sikerül csak összeszedegetni az összetartozó információelemeket.

Sajnálatosan kevés segítséget nyújt ebben a tárgymutató, amely tárgyában is, mutatójában is roppant hiányos. (A heappel kapcsolatos dolgokat éppúgy kifelejti, mint a menüelemek aktivizálására szolgáló wm_Command és wm_SysCommand üzeneteket, az utalások közül nemegyszer éppen a legfontosabb hiányzik, pl. az ICON esetében az ikonokra való hivatkozás módja, az input fókusznál pontosan a magyarázat, stb.)

Talán a legkevésbé átgondolt, leggyengébben kidolgozott rész az egész könyvben a dialógusablakokkal foglalkozó fejezet. Amellett, hogy itt érvényesül leginkább a „kásahegyeffektus", még az olvasó logikai felfogóképességét is próbára teszik a szerzők, mindjárt a fejezet legelején. „A modális dialógusablak azt jelenti, hogy használatakor a felhasználó nem válthat át egy másik ablakra, dialógusablakra vagy menüre, amíg az aktív dialógusablak feldolgozása be nem fejeződött" — szögezik le, majd néhány sorral alább kiderül, hogy mégis átválthat egy másik Windows-alkalmazásra, csak a rendszermodális dialógusablakok esetében nem. De furcsa logika uralkodik az alfejezetekre bontásban és az anyag elrendezésében is: miután így világosan definiálták a modális dialógusablakot, a nem modálisról



azt sem mondják meg, hogy mi fán terem, csak 13 alfejezettel később. A modális és nem modális dialógusablak érdemi szembeállítására később azután mégiscsak sor kerül. Nem lehetett volna ugyanezt e kettős felosztás ismertetése után közvetlenül megtenni?

Annál kevésbé érthető mindez, mert az irodalomjegyzék tanúsága szerint ugyanezek a szerzők ugyanerről a témáról néhány évvel ezelőtt már publikáltak egy könyvet.

E hibák ellenére a könyv rengeteg értékes információt tartalmaz — csak ki kell belőle bányászni. Komoly segítséget jelenthetnek a megértésben a lemezmellékleten található példák. Talán ha az anyag kifejtésében is mindenütt a konkréttól haladnának a szerzők az absztraktabb dolgok felé, könnyebben át tudnák adni tudásukat. Ha például nem véletlenül kellene rájönnie az olvasónak arra, hogy a Windows használata közben szerzett ismeretei (pl. a kiválasztásban a TAB billentyű használata) egyes vezérlőtípusok előre definiált ablakstílusával rímelnek...

Vargha Dénes

A verseny haszonélvezői

Családregény — fejezetekben

Válogatásunkban ezúttal olyan hírekre koncentráltunk, amelyekben a konkrét termék mindig túlmutat önmagán: hol egy termékcsaládot hoz közelebb hozzánk valamelyik családtag „ügyes-bajos" dolgait állítva a kirakatba, hol pedig tartós trendek vagy új tendenciák megtestesítőjeként érdemel figyelmet.

Egy CAD/CAM-űr betöltése

A CAD/CAM piacon a hardvergyártók új fogásokkal igyekeznek megőrizni piaci pozícióikat az éles versenyben. S mint már annyirszor, akinek ez leginkább a javára válik (a nevető harmadik), ebben az esetben is a felhasználó. A mérnöki munkaállomásokat gyártó cégek ugyanis egyre-másra jelentik be régebben méregdrágán kínált gépeik olcsóbb, mégis nagy teljesítményű változatait. Tehát megindult a mozgás a felső kategória felől a szélesebb kör számára elérhető középkategória felé. Még erőteljesebb a másik irányból érkező technológiai támadás: a PC ma már sokkal többet tud, mint évekkel ezelőtt, és számos gyártó tűzte ki célul a mérnöki alkalmazások területére való belépést.

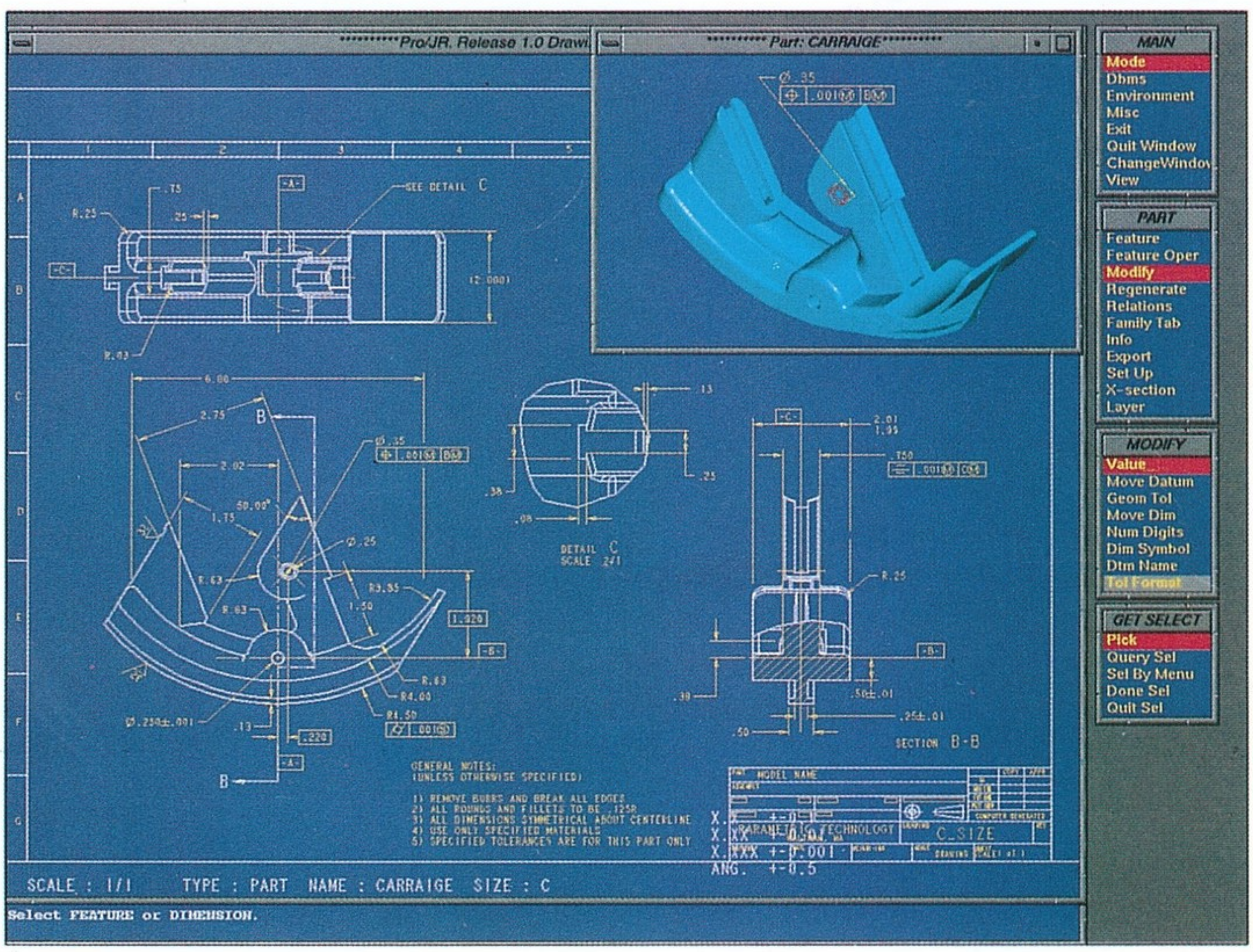
Hasonló tendenciákat figyelhetünk meg a mérnöki szoftverek kínálatában is. A számítástechnikai hőskorban a mainframe gépeken futó szoftvereknek nem volt alternatívájuk. Az egy munkahelyre jutó költségek 80-150 ezer dolláros intervallumban mozogtak. Azután sorra jelentek meg az új szoftverek, amelyek teljesítménye eleinte ugyan elmaradt a nagy elődökétől, de áruk azokénak csak a töredéke volt, így sokkal több helyen kerülhettek a mérnökök asztalára.

Időközben két — jól elkülöníthető — irányzatra szakadt a CAD/CAM piac. A mainframe gépekre fejlesztett szoftverek mellett jelentkeztek a már munkaállomásokon futó, korszerűbb technológiát képviselő szoftvergenerációk. Ezekből formálódott meg az ún. high-end („felvégi") kategória, benne előbb a 3D felületmodellezéssel, majd a testmodellezéssel. A szoftvergyártók komplex, bonyolult feladatok megoldását kínálták, amelyeknek egy munkahelyre jutó együttes hardver- és szoftverköltsége átlagosan 40-80 ezer dollár között mozgott.

A másik nagy kategória az ún. low-end („alvégi") szoftverek köre, amelybe zömmel az olcsó, 2 dimenziós rajzolóeszközök tartoznak. Itt egy komplett munkahely 5-10 ezer dolláros költséggel alakítható ki.

A piaci felmérések mutattak rá, hogy az egymástól elég távol elhelyezkedő két kategória közötti űr kihasználható lenne, mert van kereslet a kettő között valahol félúton elhelyezkedő termékegyüttes iránt is. Erre a kihívásra ad választ a napjainkban megszülető, ún. midrange (középkategória), amelyben egy munkahely költsége a 15-30 ezer dolláros tartományban helyezkedik el. A szakértők ennek a potenciális piacnak a nagyságát 600 ezer és 1 millió közötti (!) számú munkahelyre taksálják. Az ipari alkalmazások átrendeződésére lehet tehát számítani.

A fenti világtendenciára annak kapcsán érdemes felfigyelni, hogy a legdinamikusabban fejlődő CAD/CAM cég, a Parametric Technology Budapesten is bemutatta „legifjabb" CAD/CAM rendszerét, a Pro/Juniort. A grafikus Unix munkaállomással együtt (Indy R4600) kínált konfigurációt Magyarországon 400-500 ezer forintos kedvezménnyel kínálják, így az 2 millió forintnál kevesebbe, illetve más felszereltséggel valamivel 2 millió feletti összegbe kerül.



A „vevőfigyelő" Compaq és IBM

Úgy tűnik, a minőségi hardvergyártás két éllovasa fél szemét mindig a másikon tartja. Az esetek zömében csak hetek vagy napok választják el egymástól bejelentéseiket: hol a Compaq lép előbb, hol az IBM, és a válasz ritkán késik. Mindkét cég a CeBIT körüli időkre koncentrálta legérdekesebb tavaszi újdonságait, és az ott bejelentett termékek első szállításai napjainkban érkeznek meg Magyarországra is.

Amint egyik korábbi számunkban már jeleztük, teljességgel megújult a Compaq ProLinea és Deskpro családja. Határozott egységesítési törekvést figyelhetünk meg, hiszen a két család csak az integrált hálózati funkcióban különbözik egymástól, kialakításuk egyébként hasonló: Triflex/PCI architektúra, PCI helyi busz, a Deskprót azonban integrált hálózati vezérlőkkel is felruházták.

Az új stratégiával a Compaq könnyebbé kívánja tenni a hálózatban működtetett asztali PC-k kezelését, egyszerre igyekszik nagyobb hangsúlyt adni a hálózatüzemeltetésnek, hozzájárulni az alkalmazási költségek csökkentéséhez, és fokozni a PC-k teljesítményét. A mintegy ötven modellt felölelő két gépcsaládnál néhány formatervezési változtatás az alacsonyabb gyártási költséget és az egyszerűbb szervizt szolgálja.

A Compaq az intelligens kezelhetőség (intelligent manageability) kiterjesztése érdekében hirdette meg DMSPP tervet, amelyhez már a PC-s LAN-ok több vezető forgalmazója is csatlakozott. Három fő feladatról van szó: az erőforrás-ellenőrző (AssetControl) végzi a PC hardver- és szoftvereszközeinek hatékony számbavételét és ellenőrzését; a hibakezelő igyekszik minimálisra szorítani az állásidőt; a biz-



tonsági felügyelet a jogosulatlan hozzáférés megakadályozására törekszik.

Az árakról: az új, nagy teljesítményű ProLinea 575 (420 Mbájtos merevlemez, 8 Mbájt RAM, DOS/Windows) mintegy 2200 dollárnak megfelelő forintösszegbe kerül majd, az ugyanilyen konfigurációjú Deskpro 575-ös (beépített NIC-vel) várható alapára 2350 dollár.

Még egy kategóriában tartogatott meglepetést a Compaq. A ProLiant 1500-as szervercsaládot toldotta meg egy kiegészítéssel: az 5/120-as név az Intel 120 MHz-es Pentiumának integrálását takarja, de ez az architektúra már a később kifejlesztendő processzorokra való átállást is lehetővé teszi. Ára valahol 11 ezer dollár körül alakul.

Az IBM tavaszi újdonságai közül két kategória kívánkozik reflektorfénybe (méretükben igencsak eltérnek egymástól). Az innovatív megoldású, 701C betűjelű Thinkpad — népszerű nevén Butterfly — már előző számunk CeBIT-beszámolójában szerepelt. Az IBM szerint ez a termék újradefiniálhatja a szupernotebook kategóriát, mert kis mérete ellenére valódi noteszgép-szolgáltatásokat nyújt. Ehhez a gyártók ihlető forrása a felhasználók kívánságlistája volt. A felhasználó ugyanis telhetetlen: elvárja, hogy gépének „üzemelő" része minél kisebb, minél könnyebb legyen, ugyanakkor pedig a kezelési felület — a képernyő, a billentyűzet — igazodjék az ergonómiai normákhoz, az ember fizikai adottságaihoz. Magyarán azt szeretné, hogy a mobil eszközök használata ugyanolyan kényelmes legyen, mint az asztali gépeké. Ezt az elsőre „fából vaskarika" dilemmát az IBM a 701C-vel felerészt már megoldotta. A megnagyított, szétterülő, puzzle megoldású billentyűzet (85 darab 19 mm-es nyomógomb, beépített numerikus keypad, 12 darab funkcióbillentyű, TrackPoint III) után most kíváncsian várjuk, hogy mikor — és főleg hogyan! — oldják majd meg a „felfújt" képernyő problémáját.

Egyébként a 701-es család néhány paramétere:

— DX4/75 MHz, 16 K belső cache, 8—24 MB memória, vagy DX2/50 MHz, 8 K belső cache, 4—20 MB memória

— 10,4 inches aktív mátrix TFT vagy dual scan STN kijelző

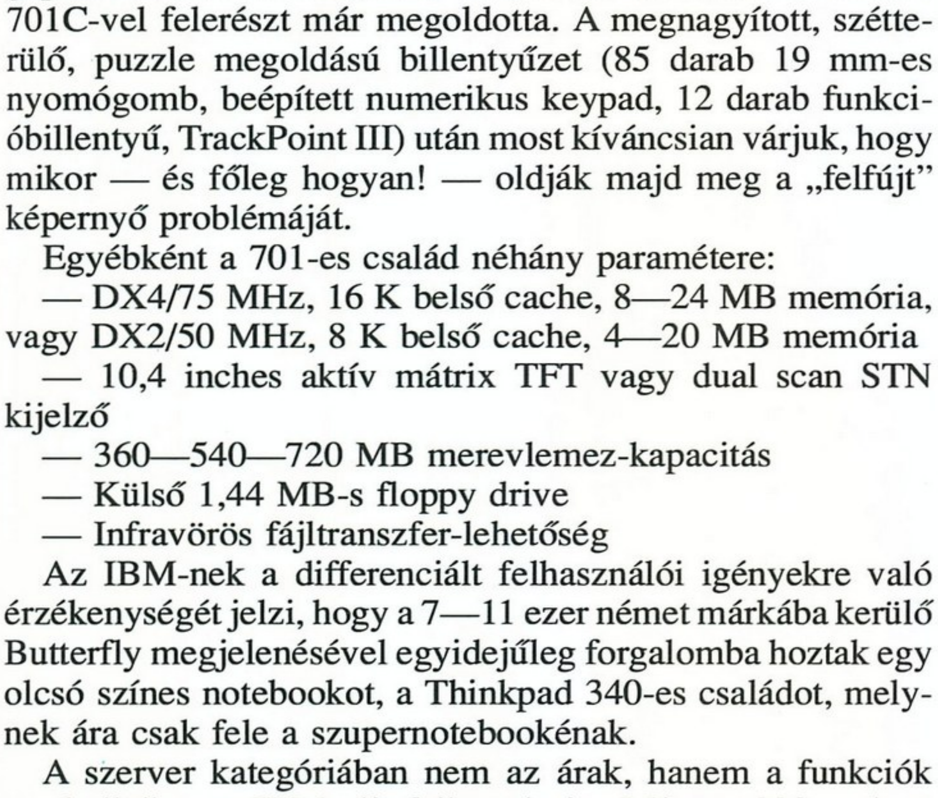
— 360—540—720 MB merevlemez-kapacitás

— Külső 1,44 MB-s floppy drive

— Infravörös fájltranszfer-lehetőség

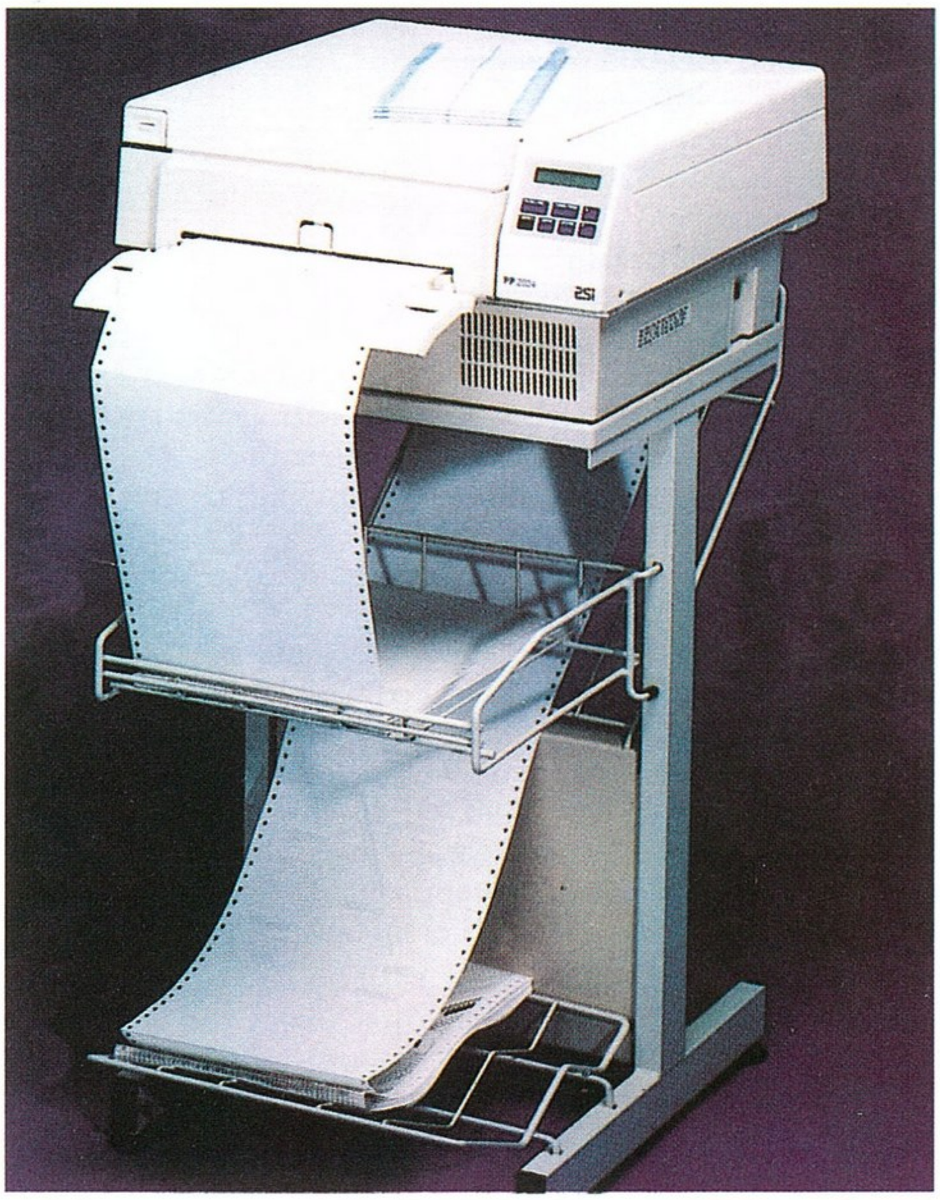
Az IBM-nek a differenciált felhasználói igényekre való érzékenységét jelzi, hogy a 7—11 ezer német márkába kerülő Butterfly megjelenésével egyidejűleg forgalomba hoztak egy olcsó színes notebookot, a Thinkpad 340-es családot, melynek ára csak fele a szupernotebookénak.

A szerver kategóriában nem az árak, hanem a funkciók motiválták az IBM új fejlesztéseit. Míg a 320-asok a 300-asokhoz képest 1 processzorhellyel (Pentium 54-essel) bővültek, és a kisvállalati igényekhez igazodnak, a 720-asnál már 6 darab 100-as Pentium számára van hely. Ez utóbbi az IBM legfejlettebb multiprocesszoros, nagy- és középvállalatoknak való hálózati szervere, amelyben az alaplapra került a menedzsmentrész, az ECC memória 1 Gbájtig bővíthető, az adatátvitel sebessége pedig 132 Mbit/s.



Nyomtatás minden mennyiségben

A nyomtatott anyag, a „hardcopy" igénye egyidős a számítástechnikai feladatok megoldásával, joggal tételezhető fel tehát, hogy a nyomtatók népes családján belül már minden feladatra kínál a piac sokféle megoldást. Meglepőnek tűnhet tehát, hogy az 1992-ben alapított PSI talált egy olyan terméktartományt, amelyben úttörőnek számít: a nagy teljesítményű mátrixnyomtatók (Epson DFX 5000) és a legkisebb kategóriájú Mannesmann-Tally sornyomtatók közötti rést. E kategória elsősorban a nagyobb (10 felhasználó feletti) hálózatok nyomtatási feladatait hivatott ellátni.



A PSI szakemberei a Philipsből váltak ki, több évtizedes tapasztalatra tettek szert a precíz mechanikák területén, és már mintegy 150 000 nyomtatót installáltak. A PSI magyarországi disztribúciós feladatait egy dinamikusan fejlődő fiatal hazai cég, a Nádor Rendszerház látja el (ahol a „vevő a király"). A Nádor forgalmának mintegy 80%-a származik a nagy tételben beszerzőktől, így a termékskála kialakításánál is elsősorban az ő igényeikre figyelnek. A nagyfelhasználók kiszolgálásánál középponti kérdés, hogy a nyomtatás legyen gyors, jó minőségű és megbízható. A Nádor kínálatában megtalálható az 1400 sor/perces Mannesmann-Tally sor-



nyomtató, amely a maga műfajában ma is világcsúcstartó, csakúgy, mint a lényegében ipari szabványú HP lézernyomtatók a magukéban. A Risograph 130 oldal/perces sokszorosítási teljesítményével számítógéphez kapcsolva ugyancsak alkalmas nyomtatási feladatok rugalmas és gyors megoldására.

A PSI céggel kötött disztribútori szerződést a CeBIT-en szentesítették, s a hiánypótló kategória mellett a csúcsteljesítményű lézernyomtatók is megjelennek kínálatukban. A PP402 jelzésű 24 tűs munkahelyi nyomtatók 230 karakter/s sebességre képesek, végfelhasználói áruk 83 900 forinttól 91 900 forintig terjed. Az 500 oldal/óra (20 000 oldal/hó) kapacitású PP405-ös, 24 tűs irodai nyomtató 499 000 forintba kerül, míg a 408-as jelzésű (750 oldal/óra — 30 000 oldal/hó) nyomtató 830 000 forintért szerezhető be. A 300x300 dpi-s, 30 oldal/perces lézernyomtató ára már 3 millió felett van, ugyanennek duplex kivitelű változata pedig 4 234 900 forintot tesz ki.

Többek között a PSI-nyomtatók meghonosításával a nyomtatási sebességek egymáshoz viszonyításában bizonyára sokan megbarátkozunk az oldal/óra mutatóval, amely sok szempontból pontosabb képet ad, mint a megszokott karakter/s. A nagyobb cégeknek mindenesetre így egyszerűbb kiszámítani, hogy hány ezer nyomtatott oldallal kalkulálhatnak egy hónapban.

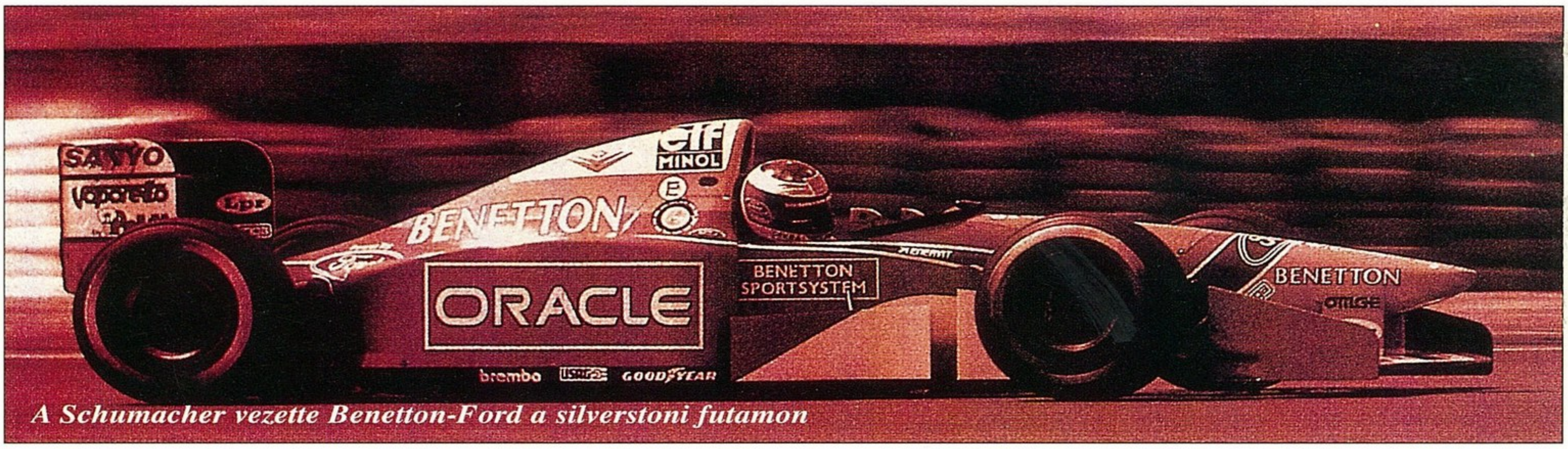
S ha már itt tartunk, figyelmet érdemelnek a Magyarországon is gyorsan terjedő OKI-nyomtatók újdonságai. Ilyen a mátrixnyomtatók családjában a megújított Microline 500-as sorozat. A program keretében a 9 tűs modelleket — ML 520/1 Elite — színes opcióval látják el, a nagy volumenű irodai felhasználás céljainak és igényeinek megfelelően. E kategóriában nóvumnak számít a papíradagolás megoldása, ami főleg azoknak jelent gyakorlati előnyt, akik munkájuk során több különböző formátumú papírt felváltva használnak. Az OKI a nyomtatófej élettartamát 200 millió karakterben, a festékszalagét 4 millió karakterben jelöli meg.

Gazdaságos és üdvözlendő megoldás a percenként 4 lapot kibocsátó nyomtatóknál az Adobe Memory Booster integrálása, amelynek révén az OL 400ex és OL 410ex nyomtatók tulajdonosai a printer memóriájának bővítése nélkül is élhetnek a PostScript II kínálta előnyökkel.

A 600 dpi-s oldalnyomtatók kategóriájában a workgroup-alkalmazásokhoz kínált OL 1200ex-et annak hálózati tudása, kellemes papíradagolása és 12 lap/perces sebessége teheti vonzóvá, az OL 810ex-nél a 8 lap/perces sebesség biztosítására 25 MHz-es RISC processzor és 32 bites adatsín szolgál, az OKI's Microfine Spherical Toner pedig szintén hozzájárul ahhoz, hogy a nyomat minősége közelítsen a fotóminőséghez; az élek, a vonalak, a sarkok, a fekete tónusok „hajszálpontosan" jelenjenek meg.

Varga János

Ez a lapszám
nem jelenhetett volna meg határidőre,
ha a Kürt Kft nem javította volna meg azonnal
központi gépünk alap(lap)os hardverhibáit!



A Schumacher vezette Benetton-Ford a silverstoni futamon

JÓ FORMÁBAN AZ ORACLE!

Győzni a legjobbak között! A nagy álmot legutóbb a Schumacher vezette Benetton-Ford Forma 1-es csapat váltotta valóra. A kiélezett küzdelemben a siker elérése csak az ember és a technika tökéletes összhangjával lehetséges. A csúcsfordulaton pörgő verseny állandó fejlődést, fejlesztést követel a konstruktőröktől, versenyzőtől, csapattól egyaránt.

Nagy tömegű információ feldolgozására, számtalan lehetséges variáció kipróbálására van szükség minden másodpercnyi teljesítményjavulás eléréséhez.

A versenyautók világa szűkebb környezetben jelenti azt az információkezelési közeget, ahol a vállalatokhoz hasonlóan soktényezős folyamatok egyidejű menedzselése szükséges. Ezért találkozott az Oracle és Benetton csapat egymással. Szemléletükben, működésükben sok a közös vonás. A Benettonnál a versenygépek tervezése, megvalósítása, hangolása, a versenyeken való részvétel, a fantasztikus kerékcserék, a külső szállítókkal, motor- és gumiköpeny-gyártókkal folytatott együttműködés olajozottan, összehangolt teammunkában történik.

Hasonló a helyzet az ORACLE cégnél is, ahol a saját működés, illetve a felhasználótámogatás és -kiszolgálás a teammunkán alapul. A másik hasonlóság a személyes szakértelem fontossága. A Forma 1-es versenyzők rátermettségének, tehetségének szükségessége alapvető. Emellett az egyén, a versenyző a csapatmunkának is részese, együtt él a problémákkal, a feladatokkal. Észrevételeivel folyamatosan segíti a konstruktőrök, technikusok, a műszaki stáb munkáját. Így vesznek részt az ORACLE felkészült, gyakorlott szakemberei is a cég jól szervezett, sikeres csapatmunkájában. A Benetton és az ORACLE működésében megtalálható hasonlóságok közül még néhány példa: a magas minőségre való törekvés, a tevékenység nemzetközi jellege, a kihívások, feladatok mértéke, az élmezőnyhöz tartozás és az elért győzelmek meghatározó szerepe, stb.

Nem véletlenek a felsorolt azonosságok, hiszen az ORACLE Corporation – a világ 93 országában működő, kaliforniai szoftvercég – a nemzetközi számítástechnikai piac meghatározó szereplője. Adatbázis kezelő rendszerek fejlesztőjeként indult és ma is ez az egyik fő profilja. Az elsők között szerepel az adatbázis kezelők és a nyílt rendszerek fejlesztésében. Ugyanakkor a kínálatban egyre nagyobb helyet kapnak a kész alkalmazási rendszerek. Az ORACLE évek óta dinamikusan fejlődik. Éves növekedési rátája eléri a 35 százalékot. Fő törekvése, hogy relációs adatbázis kezelői és fejlesztőeszközei folyamatos korszerűsítése mellett egyre nagyobb mértékben kínáljon komplett megoldásokat, rendszerintegrációt felhasználóinak.

Mindezekkel összhangban az ORACLE jelentős fejlődést kíván elérni az elkövetkező néhány évben. Növeli alkalmazottainak számát, megtöbbszörözi bevételeit.

A Magyarországon működő ORACLE Hungary szintén jó eredményeket ért el és tervezi szervezetének, tevékenységének hasonló továbbfejlesztését.

ORACLE szoftverrendszereket sikerrel alkalmaznak már országszerte. Az ORACLE Hungary ugyanazokat a magas minőségi termékeket, ugyanazt a kiterjedt szolgáltatásrendszert nyújtja felhasználóinak, mint az ORACLE bárhol a világon.

Győződjön meg róla Ön is!

ORACLE®

ORACLE HUNGARY

1149 Budapest, Egressy út 20.
Tel.:251-4100; Fax: 251-4108

IFABO A/113B

INFORMÁCIÓKÉRÉS: 0546 ◄